



KERJA PRAKTIK - IF184801

Rancang Bangun Aplikasi Portal Manajemen Ujian Sekolah SMA Jawa Timur 2020 Menggunakan Kerangka Kerja Laravel

Dinas Pendidikan Provinsi Jawa Timur

Jl. Genteng Kali No.33, Genteng, Kec. Genteng, Kota SBY, Jawa Timur 60275

Periode: 16 Desember 2019 - 23 Maret 2020

Oleh:

Muhammad Jaya Siraj

05111740000088

Pembimbing Jurusan

Yudhi Purwananto S.Kom., M.Kom.

Pembimbing Lapangan

Dwi Sunaryono, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Rancang Bangun Aplikasi Portal Manajemen Ujian Sekolah SMA Jawa Timur 2020 Menggunakan Kerangka Kerja Laravel

Oleh:

Muhammad Jaya Siraj

05111740000088

Disetujui oleh Pembimbing Kerja Praktik:

1. Yudhi Purwananto S.Kom., M.Kom.
NIP. 19700714 199703 1 002

.....
(Pembimbing Departemen)

2. Dwi Sunaryono, S.Kom., M.Kom.
NIP. 19720528 199702 1 001

.....
(DWI SUNARYONO S.Kom., M.Kom.)
NIP/NIK 1972.0528199702.100.
(Pembimbing Lapangan)

**SURABAYA
Desember, 2020**

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
DAFTAR ISI.....	iv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Maksud dan Tujuan	2
1.3 Sasaran	3
1.4 Ruang Lingkup	3
1.5 Ikhtisar Dokumen	5
BAB 2 DESKRIPSI UMUM APLIKASI	8
2.1. Deskripsi Umum Sistem.....	8
2.2. Karakteristik Pengguna	9
2.3. Lingkungan Operasi	12
2.4. Batasan Perancangan dan Implementasi Sistem.....	12
BAB 3 PROSES BISNIS APLIKASI	14
3.1 Proses 1	14
3.2 Proses 2	15
BAB 4 PERANCANGAN DATA MODEL	16
4.1 Data Model Aplikasi Portal Manajemen USP-BKS....	16
4.1.1 Tabel encryption_keys	17
4.1.2 Tabel jadwal_ujian	17
4.1.3 Tabel kab_kota	18
4.1.4 Tabel log_sn	19
4.1.5 Tabel Paket	20

4.1.6	Tabel pelajaran	20
4.1.7	Tabel sekolah.....	21
4.1.8	Tabel serial_numbers	22
4.1.9	Tabel settings.....	22
4.1.10	Tabel soal	23
4.1.11	Tabel ujian_paket	24
4.1.12	Tabel ujian_siswa	25
4.1.13	Tabel user	26
BAB 5 USER GUIDE PROKTOR		29
5.1.	Halaman Utama Portal Proktor	29
5.2.	Menu Download Data	32
5.3.	Menu Request Server	34
5.4.	Menu Ubah Password Portal	35
5.5.	Menu Download Soal Esai	36
BAB 6 USER GUIDE ADMINISTRATOR		38
6.1.	Halaman Utama Portal Proktor	38
6.2.	Menu Log Update Jadwal.....	43
6.3.	Menu Log Admin	43
6.4.	Menu Monitoring Download.....	44
6.5.	Menu Monitoring SN dan Server	45
6.6.	Menu Monitoring Hasil Ujian	47
6.7.	Menu Manajemen SN dan Server.....	49
6.8.	Menu Akses Download Kabupaten/Kota	52
6.9.	Menu Manajemen Ujian	53
6.10.	Menu Manajemen Paket-Soal.....	59

6.11.	Menu Manajemen Sekolah	63
6.12.	Menu Manajemen User	66
6.13.	Menu Manajemen Pelajaran	69
6.14.	Menu Manajemen Siswa	72
BAB 7 IMPLEMENTASI SISTEM.....		75
7.1.	Model	75
7.2.	Controller	91

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Ujian sekolah yang dilakukan oleh SMA Negeri merupakan salah satu kebutuhan tahunan pada dunia pendidikan. Dengan banyaknya SMA Negeri yang ada di Jawa Timur, sehingga Dinas Pendidikan Provinsi Jawa Timur membuat ujian sekolah dengan sistem *online*. Beberapa tahun terakhir sudah diterapkan pelaksanaan ujian sekolah dengan sistem *online* tersebut dan dilaksanakan secara serentak pada Provinsi Jawa Timur. Pada tahun ajaran 2019/2020 ujian sekolah berbasis *online* tersebut mengalami perkembangan. Perkembangan tersebut adalah ujian tidak hanya dapat dilakukan pada komputer, akan tetapi ujian juga dapat dilakukan dengan menggunakan *smartphone*. Sehingga pada tahun ajaran 2019/2020, Ujian Sekolah Berbasis *Online* berganti nama menjadi Ujian Satuan Pendidikan Berbasis Komputer dan *Smartphone* (USP-BKS). Untuk memudahkan dalam proses pemantauan selama USP-BKS berlangsung, maka dibangun sebuah Aplikasi Portal Manajemen Ujian

Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS).

1.2 Maksud dan Tujuan

Dokumen ini berisi spesifikasi Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) SMA Negeri Tahun Pelajaran 2019/2020 di Wilayah Jawa Timur. Maksud dari pekerjaan ini adalah untuk melakukan penyusunan aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) SMA Negeri Provinsi Jawa Timur Tahun Pelajaran 2019/2020.

Tujuan dari pekerjaan ini yaitu mendukung proses pemantauan ujian sekolah SMA Negeri Provinsi Jawa Timur pada tahun pelajaran 2019/2020 dengan menggunakan Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS). Sedangkan tujuan dari penulisan dokumen ini adalah untuk memberikan penjelasan mengenai perangkat lunak yang akan dibangun baik berupa gambaran umum maupun penjelasan detail dan menyeluruh.

1.3 Sasaran

Sasaran dari pekerjaan ini yaitu menyelesaikan Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) SMA Negeri pada tahun pelajaran 2019/2020 di Provinsi Jawa Timur dengan pelayanan yang baik dan sesuai dengan kebutuhan. Sedangkan sasaran dari dokumen ini adalah akan digunakan sebagai bahan acuan dalam proses pengembangan dan sebagai bahan evaluasi pada saat proses pengembangan perangkat lunak maupun di akhir pengembangannya. Dengan adanya dokumen ini diharapkan pengembangan perangkat lunak akan lebih terarah dan lebih terfokus serta tidak menimbulkan ambiguitas terutama bagi pengembang perangkat lunak sistem informasi.

1.4 Ruang Lingkup

Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) merupakan aplikasi untuk manajemen dan monitoring ujian sekolah siswa SMA Negeri di wilayah Jawa Timur. Sistem ini dapat memfasilitasi dua jenis pengguna dengan lebih mudah.

Pengguna sistem informasi antara lain proktor, dan administrator sistem. Dengan adanya Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) diharapkan kegiatan ujian siswa SMA Negeri menjadi lebih terstruktur dan dapat dimonitoring dengan baik. Pada Aplikasi Portal Manajemen USP-BKS ini mencakup beberapa fungsi utama, antara lain:

- Fungsi Monitoring Jadwal
- Fungsi Monitoring Admin
- Fungsi Monitoring Download
- Fungsi Monitoring SN dan Server
- Fungsi Monitoring Hasil Ujian
- Fungsi Manajemen SN dan Server
- Fungsi Manajemen Akses Download Kabupaten/Kota
- Fungsi Manajemen Ujian
- Fungsi Manajemen Paket Soal
- Fungsi Manajemen Sekolah
- Fungsi Manajemen User
- Fungsi Manajemen Mata Pelajaran
- Fungsi Manajemen Peserta Ujian / Siswa

- Fungsi Impor Data Pelajaran, Sekolah, Siswa, Data Ujian dan Soal
- Fungsi Unduh dan Cetak Berita Acara, Daftar Hadir, dan Kartu Peserta
- Fungsi Koreksi dan Lihat Hasil Ujian

1.5 Ikhtisar Dokumen

Dokumen ini secara garis besar terdiri dari 6 bab dengan perincian sebagai berikut :

- **Bab 1 Pendahuluan**, merupakan pengantar dokumen yang berisi latar belakang penulisan dokumen, maksud dan tujuan dari pengerjaan aplikasi dan penulisan dokumen, sasaran aplikasi dan dokumen, ruang lingkup pengerjaan aplikasi, serta deskripsi umum dokumen yang merupakan ikhtisar dokumen.
- **Bab 2 Deskripsi Umum Aplikasi**, yang berisi deskripsi umum sistem, karakteristik pengguna, lingkungan operasi, serta batasan perancangan dan implementasi sistem.

- **Bab 3 Proses Bisnis Aplikasi**, mendeskripsikan proses-proses bisnis yang akan dilakukan pada Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS).
- **Bab 4 Perancangan Data Model**, yang mendeskripsikan model data yang digunakan dalam pembuatan Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS).
- **Bab 5 User Guide Proktor**, yang menjelaskan cara penggunaan dari Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) untuk proktor.
- **Bab 6 User Guide Administrator**, yang menjelaskan cara penggunaan dari Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) untuk administrator.
- **Bab 7 Implementasi Sistem**, yang berisi code yang digunakan dalam membangun Aplikasi

Portal Manajemen Ujian Satuan Pendidikan
Berbasis Komputer dan Smartphone (USP-BKS).

BAB 2

DESKRIPSI UMUM APLIKASI

2.1. Deskripsi Umum Sistem

Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) merupakan sebuah perangkat lunak sistem informasi berbasis website yang digunakan untuk memenuhi segala bentuk proses bisnis yang berhubungan dengan aktivitas manajemen dan monitoring ujian siswa SMA Negeri di wilayah Jawa Timur sehingga mempermudah dalam memantau dan mengatur jalannya ujian sekolah *online* tersebut.

Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) digunakan oleh 2 pengguna yaitu proktor sebagai pengguna yang melakukan manajemen terhadap ujian yang berlangsung, dan administrator sebagai pengguna yang memiliki hak akses penuh terhadap sistem. Dimana administrator dapat melakukan manajemen sekolah, user, pelajaran, siswa, ujian, paket soal, monitorin ujian, dan lain sebagainya. Aplikasi ini digunakan oleh pengguna yang sudah memiliki hak akses

terhadap Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS).

2.2. Karakteristik Pengguna

Karakteristik pengguna dijabarkan dalam tabel berikut ini:

Tabel 2 Karakteristik Pengguna

Kategori Pengguna	Tugas	Hak Akses ke Aplikasi
Proktor	<ul style="list-style-type: none"> - Melakukan manajemen ujian sekolah 	<ul style="list-style-type: none"> - Proktor dapat melakukan download data ujian - Proktor dapat melakukan download data siswa - Proktor dapat melakukan <i>request server</i> - Proktor dapat mengubah password portal - Proktor dapat mengunduh soal essai

Administrator	<ul style="list-style-type: none"> - Melakukan manajemen dan monitoring Aplikasi Ujian Sekolah 	<ul style="list-style-type: none"> - Administrator dapat melihat log update jadwal - Administrator dapat melihat log admin - Administrator dapat melakukan monitoring download - Administrator dapat melakukan monitoring SN dan Server - Administrator dapat melakukan monitoring hasil ujian - Administrator dapat melakukan manajemen SN dan Server - Administrator dapat mengatur akses
---------------	---	--

		<p>download kabupaten/ kota</p> <ul style="list-style-type: none"> - Administrator dapat melakukan manajemen ujian - Administrator dapat melakukan manajemen paket soal - Administrator dapat melakukan manajemen sekolah - Administrator dapat melakukan manajemen pengguna - Administrator dapat melakukan manajemen pelajaran - Administrator dapat melakukan manajemen siswa
--	--	--

2.3. Lingkungan Operasi

Lingkungan operasi untuk menjalankan Aplikasi USP-BKS ini adalah sebagai berikut :

1. Aplikasi USP-BKS hanya dapat dijalankan pada sistem operasi Windows 7 SP1 yang telah memiliki VHD.
2. Komputer yang digunakan memiliki ruang kosong pada *harddisk* minimal 40GB dan *memory* 2GB.
3. Aplikasi USP-BKS hanya dapat dijalankan pada smarphone Android yang sudah terinstall exam browser yang sudah disediakan dengan sistem operasi minimal Android 5
4. Aplikasi USP-BKS dapat dijalankan di komputer dan smartphome dengan browser yang tersambung dengan internet

2.4. Batasan Perancangan dan Implementasi Sistem

Pengembangan Aplikasi USP-BKS ini memiliki keterbatasan-keterbatasan yaitu sebagai berikut :

1. Aplikasi USP-BKS dibuat dengan menggunakan bahasa PHP, HTML, CSS, JQuery, Ajax dan Javascript.

2. Hak akses pengguna hanya dimiliki oleh pengguna yang terdaftar hak aksesnya di dalam database
3. Software pendukung yang digunakan adalah DBMS MySQL

BAB 3

PROSES BISNIS APLIKASI

Di dalam Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS) terdapat 2 hak akses yang berbeda yaitu Proktor, dan Administrator. Dimana alurnya dimulai dari Proktor yang melakukan download data pelajaran, sekolah, & siswa dan juga data jadwal & soal, Proktor juga dapat melakukan *request server*, mengubah password, dan mengunduh soal esai. Sedangkan administrator akan melakukan manajemen dan monitoring penuh terhadap aplikasi.

3.1 Proses 1

Proktor melakukan melakukan download data pelajaran, sekolah, & siswa dan juga data jadwal & soal, kemudian proktor dapat melakukan *request server* sesuai dengan jumlah server yang diperlukan pada sekolah masing-masing. Proktor juga dapat mengubah password portal agar lebih mudah diingat dan lebih aman. Kemudian proktor juga dapat mengunduh soal esai yang diujikan pada ujian untuk memudahkan dalam proses melakukan koreksi jawaban esai.

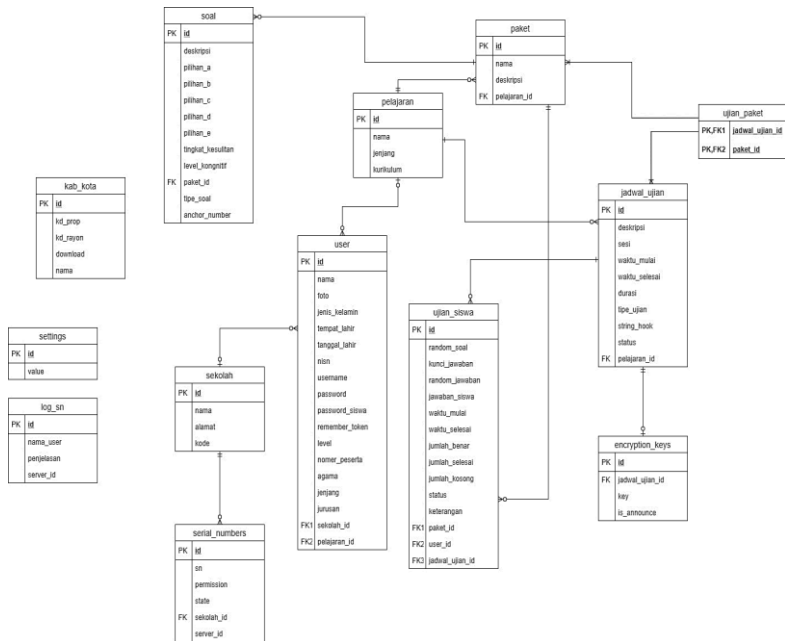
3.2 Proses 2

Administrator akan melakukan monitoring dan manajemen terhadap keseluruhan Aplikasi Portal Manajemen Ujian Satuan Pendidikan Berbasis Komputer dan Smartphone (USP-BKS). Administrator memiliki hak penuh terhadap aplikasi USP-BKS tersebut. Monitoring yang dilakukan meliputi log update jadwal, log admin, monitoring download, monitoring SN dan Server dan monitoring hasil ujian. Sedangkan pada bagian manajemen, administrator dapat melakukan manajemen SN dan Server, akses download kabupaten / kota, manajemen ujian, manajemen paket soal, manajemen sekolah, manajemen user, manajemen pelajaran dan manajemen siswa.

PERANCANGAN DATA MODEL

4.1 Data Model Aplikasi Portal Manajemen USP-BKS

Data model yang digunakan dalam Aplikasi Portal Manajemen USP-BKS ini adalah sebagai berikut:



4.1.1 Tabel encryption_keys

Tabel encryption_keys merupakan tabel yang digunakan untuk menyimpan kunci untuk membuka ujian. Tabel ini memiliki beberapa atribut, antara lain adalah id, jadwal_ujian_id, key, dan is_announce. Dimana atribut id merupakan primary key dari tabel encryption_keys. Sedangkan atribut jadwal_ujian_id merupakan foreign key dari atribut id pada tabel jadwal_ujian.

No	Atribut	Tipe Data	Panjang
1	id	int	10
2	jadwal_ujian_id	char	36
3	key	varchar	191
4	is_announce	varchar	191

4.1.2 Tabel jadwal_ujian

Tabel jadwal_ujian merupakan tabel yang digunakan untuk menyimpan data-data lengkap mengenai jadwal ujian yang akan dilakukan. Tabel ini memiliki beberapa atribut, antara lain adalah id, deskripsi, sesi, waktu_mulai, waktu_selesai, durasi, tipe_ujian, string_hook, status, dan pelajaran_id. Dimana atribut id merupakan primary key dari

tabel jadwal_ujian. Sedangkan atribut pelajaran_id merupakan foreign key dari atribut id pada tabel pelajaran.

No	Atribut	Tipe Data	Panjang
1	id	char	36
2	deskrpsi	varchar	191
3	sesi	varchar	191
4	waktu_mulai	datetime	-
5	waktu_selesai	datetime	-
6	durasi	int	11
7	tipe_ujian	varchar	191
8	string_hook	varchar	191
9	status	varchar	191
10	pelajaran_id	varchar	191

4.1.3 Tabel kab_kota

Tabel kab_kota merupakan tabel yang digunakan untuk menyimpan daftar kabupaten/kota yang akan mengikuti ujian. Tabel ini memiliki beberapa atribut, antara lain adalah id, kd_prop, kd_rayon, download, dan nama. Dimana atribut id merupakan primary key dari tabel kab_kota.

No	Atribut	Tipe Data	Panjang
1	id	bigint	20
2	kd_prop	varchar	8
3	kd_rayon	varchar	8
4	download	int	11
5	nama	varchar	50

4.1.4 Tabel log_sn

Tabel log_sn merupakan tabel yang digunakan untuk menyimpan riwayat perubahan SN yang pernah dilakukan. Tabel ini memiliki beberapa atribut, antara lain adalah id, nama_user, penjelasan, dan server_id. Dimana atribut id merupakan primary key dari tabel log_sn.

No	Atribut	Tipe Data	Panjang
1	id	bigint	20
2	nama_user	varchar	500
3	penjelasan	varchar	500
4	server_id	varchar	191

4.1.5 Tabel Paket

Tabel paket merupakan tabel yang digunakan untuk menyimpan data paket yang ada pada USP-BKS. Tabel ini memiliki beberapa atribut, antara lain adalah id, nama, deskripsi, dan pelajaran_id. Dimana atribut id merupakan primary key dari tabel paket. Sedangkan atribut pelajaran_id merupakan foreign key dari atribut id pada tabel pelajaran.

No	Atribut	Tipe Data	Panjang
1	id	char	36
2	nama	varchar	191
3	deskripsi	varchar	191
4	pelajaran_id	int	11

4.1.6 Tabel pelajaran

Tabel pelajaran merupakan tabel yang digunakan untuk menyimpan data pelajaran yang ada pada USP-BKS. Tabel ini memiliki beberapa atribut, antara lain adalah id,

nama, jenjang, dan kurikulum. Dimana atribut id merupakan primary key dari tabel pelajaran.

No	Atribut	Tipe Data	Panjang
1	id	int	10
2	nama	varchar	191
3	jenjang	varchar	191
4	kurikulum	varchar	191

4.1.7 Tabel sekolah

Tabel sekolah merupakan tabel yang digunakan untuk menyimpan data sekolah yang mengikuti USP-BKS. Tabel ini memiliki beberapa atribut, antara lain adalah id, nama, alamat, dan kode. Dimana atribut id merupakan primary key dari tabel sekolah.

No	Atribut	Tipe Data	Panjang
1	id	char	36
2	nama	varchar	191
3	alamat	varchar	191
4	kode	char	10

4.1.8 Tabel serial_numbers

Tabel serial_numbers merupakan tabel yang digunakan untuk menyimpan data serial numbers yang digunakan pada USP-BKS. Tabel ini memiliki beberapa atribut, antara lain adalah id, sn, permission, state, sekolah_id, dan server_id. Dimana atribut id merupakan primary key dari tabel serial_numbers. Sedangkan atribut sekolah_id merupakan foreign key dari atribut id pada tabel sekolah.

No	Atribut	Tipe Data	Panjang
1	id	int	10
2	sn	varchar	191
3	permission	varchar	191
4	state	int	11
5	sekolah_id	varchar	191
6	server_id	varchar	191

4.1.9 Tabel settings

Tabel settings merupakan tabel yang digunakan untuk menyimpan peraturan yang digunakan pada USP-BKS seperti batas input soal dan lain sebagainya. Tabel ini

memiliki beberapa atribut, antara lain adalah id, dan value. Dimana atribut id merupakan primary key dari tabel settings.

No	Atribut	Tipe Data	Panjang
1	id	varchar	191
2	value	varchar	191

4.1.10 Tabel soal

Tabel soal merupakan tabel yang digunakan untuk menyimpan data lengkap semua soal yang digunakan pada USP-BKS. Tabel ini memiliki beberapa atribut, antara lain adalah id, deskripsi, pilihan_a, pilihan_b, pilihan_c, pilihan_d, pilihan_e, tingkat_kesulitan, level_kognitif, paket_id, tipe_soal, dan anchor_number. Dimana atribut id merupakan primary key dari tabel soal. Sedangkan atribut paket_id merupakan foreign key dari atribut id pada tabel paket.

No	Atribut	Tipe Data	Panjang
1	id	char	36
2	deskripsi	longtext	-
3	pilihan_a	longtext	-

4	pilihan_b	longtext	-
5	pilihan_c	longtext	-
6	pilihan_d	longtext	-
7	pilihan_e	longtext	-
8	tingkat_kesulitan	int	11
9	level_koginitif	varchar	191
10	paket_id	char	36
11	tipe_soal	varchar	191
12	anchor_number	int	11

4.1.11 Tabel ujian_paket

Tabel ujian_paket merupakan tabel hasil relasi dari tabel paket dan tabel jadwal_ujian yang digunakan untuk menyimpan data jadwal ujian dan paket soal yang digunakan pada USP-BKS. Tabel ini memiliki 2 atribut yaitu jadwal_ujian_id dan paket_id. Kedua atribut tersebut merupakan primary key pada tabel ujian_paket. Dimana atribut jadwal_ujian_id juga merupakan foreign key dari atribut id pada tabel jadwal_ujian, dan atribut paket_id merupakan foreign key dari atribut id pada tabel paket.

No	Atribut	Tipe Data	Panjang
----	---------	-----------	---------

1	jadwal_ujian_id	char	36
2	paket_id	char	36

4.1.12 Tabel ujian_siswa

Tabel ujian_siswa merupakan tabel yang digunakan untuk menyimpan data ujian siswa yang merupakan data soal yang dikerjakan siswa beserta kunci jawaban dan jawaban siswa tersebut. Tabel ini memiliki beberapa atribut, antara lain adalah id, random_soal, kunci_jawaban, random_jawaban, jawaban_siswa, waktu_mulai, waktu_selesai, jumlah_benar, jumlah_salah, jumlah_kosong, status, keterangan, paket_id, user_id, dan jadwal_ujian_id. Dimana atribut id merupakan primary key dari tabel ujian_siswa. Sedangkan atribut paket_id, user_id, dan jadwal_ujian_id merupakan foreign key. Atribut paket_id merupakan foreign key dari atribut id pada tabel paket, atribut user_id merupakan foreign key dari atribut id pada tabel user, dan atribut jadwal_ujian_id merupakan foreign key dari atribut id pada tabel jadwal_ujian.

No	Atribut	Tipe Data	Panjang
1	id	char	36
2	random_soal	text	-
3	kunci_jawaban	text	-
4	random_jawaban	text	-
5	jawaban_siswa	text	-
6	waktu_mulai	datetime	-
7	waktu_selesai	datetime	-
8	jumlah_benar	int	11
9	jumlah_salah	int	11
10	jumlah_kosong	int	11
11	status	varchar	191
12	keterangan	varchar	191
13	paket_id	char	36
14	user_id	char	36
15	jadwal_ujian_id	char	36

4.1.13 Tabel user

Tabel user merupakan tabel yang digunakan untuk menyimpan data pengguna dengan berbagai kategori pengguna yang memiliki hak akses terhadap USP-BKS. Tabel ini memiliki beberapa atribut, antara lain adalah id, nama, foto, jenis_kelamin, tempat_lahir, tanggal_lahir, nisan,

username, password, password_siswa, remember_token, level, nomer_peserta, agama, jenjang, jurusan, sekolah_id, dan pelajaran_id. Dimana atribut id merupakan primary key dari tabel user. Sedangkan atribut sekolah_id dan pelajaran_id merupakan foreign key. Atribut sekolah_id merupakan foreign key dari atribut id pada tabel sekolah, sedangkan atribut pelajaran_id merupakan foreign key dari atribut id pada tabel pelajaran.

No	Atribut	Tipe Data	Panjang
1	id	char	36
2	nama	varchar	191
3	foto	varchar	191
4	jenis_kelamin	varchar	191
5	tempat_lahir	varchar	191
6	tanggal_lahir	varchar	191
7	nisn	varchar	191
8	username	varchar	191
9	password	varchar	191
10	password_siswa	varchar	191
11	remember_token	varchar	191
12	level	varchar	191
13	nomer_peserta	varchar	191

14	agama	varchar	191
15	jenjang	varchar	191
16	jurusan	varchar	191
17	sekolah_id	char	36
18	pelajaran_id	int	11

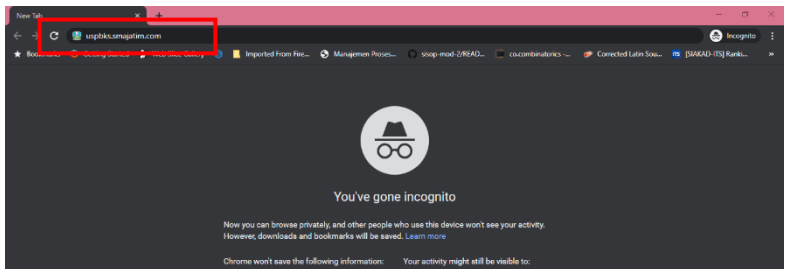
BAB 5

USER GUIDE PROKTOR

5.1. Halaman Utama Portal Proktor

Proktor merupakan pengguna aplikasi USP-BKS yang memiliki hak akses untuk melakukan manajemen dan monitoring ujian yang berlangsung. Setiap sekolah memiliki proktor masing-masing.

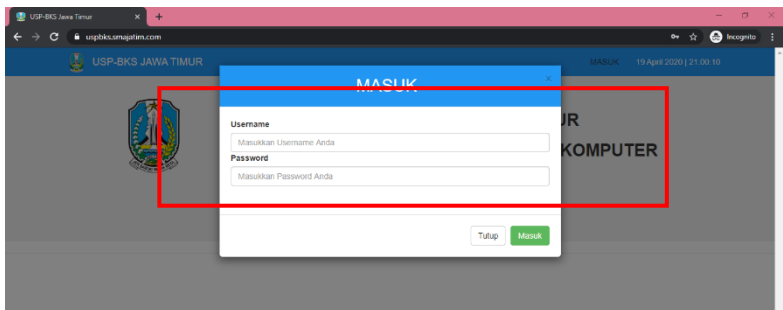
1. Proktor mengakses alamat USP-BKS yaitu `uspbks.smajatim.com` pada browser di komputer



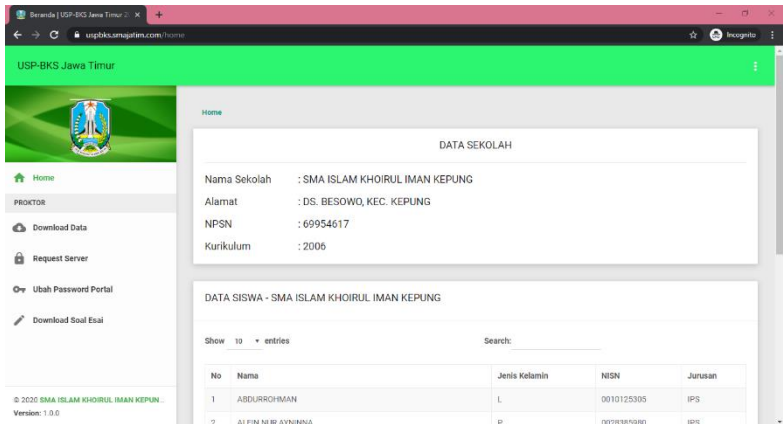
2. Maka akan muncul halaman awal dari USP-BKS



3. Proktor melakukan login dengan memilih menu **MASUK** maka akan muncul halaman dialog login. Masukkan username dan password pada form yang tersedia.



4. Setelah login, akan muncul halaman utama proktor seperti berikut.



5. Pada halaman proktor terdapat 4 menu yaitu Download Data, Request Server, Ubah Password Portal, dan Download Soal Esai.

- Download Data digunakan untuk mengunduh data pelajaran, sekolah & siswa dan juga data jadwal & soal.
- Request Server digunakan untuk melakukan request jumlah server yang akan digunakan oleh sekolah yang bersangkutan.
- Ubah Password Portal digunakan untuk mengubah password.
- Download Soal Esai digunakan untuk mengunduh soal esai yang akan digunakan pada saat ujian berlangsung.

5.2. Menu Download Data

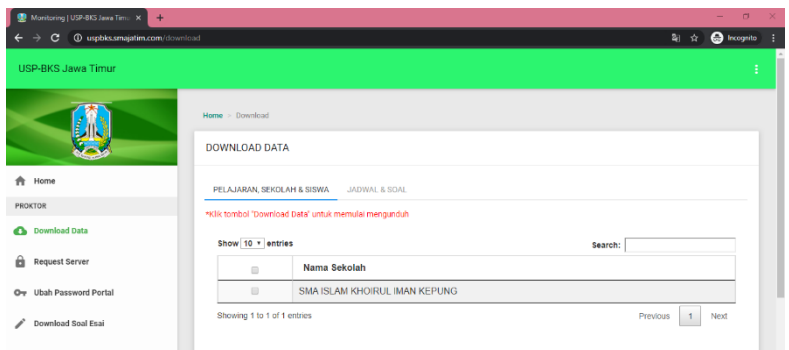
Menu download data digunakan untuk mengunduh data pelajaran, sekolah & siswa dan juga data siswa & ujian.

1. Setelah proktor melakukan login, pada halaman utama terdapat menu untuk melakukan download data. Di dalam menu tersebut terdapat 2 submenu, yaitu submenu untuk mengunduh data pelajaran, sekolah & siswa dan untuk mengunduh data jadwal & soal.



2. Untuk mengunduh data pelajaran, sekolah & siswa klik submenu

PELAJARAN, SEKOLAH & SISWA



3. Klik pada kolom untuk mencentang data sekolah yang akan di unduh

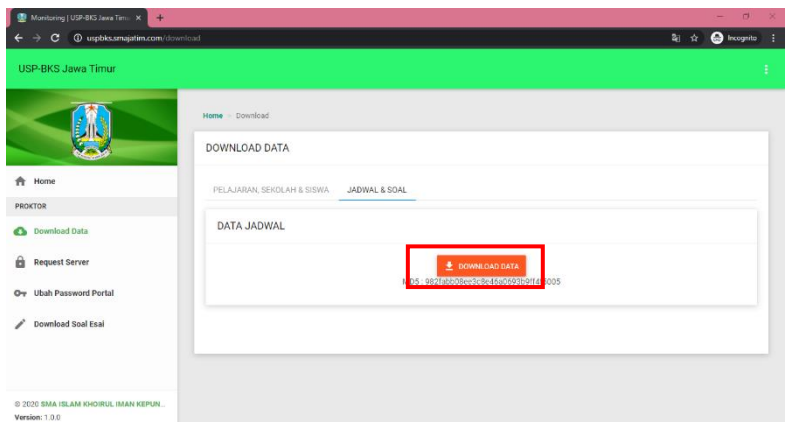


4. Klik download data



5. Untuk mengunduh data jadwal & soal klik submenu JADWAL & SOAL

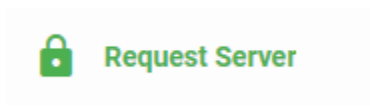
6. Klik pada tombol download data



5.3. Menu Request Server

Menu request server digunakan untuk melakukan request jumlah server yang akan digunakan pada sekolah tersebut.

1. Setelah proktor melakukan login, pada halaman utama terdapat menu untuk melakukan request server.



2. Untuk melakukan request server, masukkan jumlah server yang dibutuhkan pada kolom yang tersedia. Kemudian klik submit.

A screenshot of a web browser showing the 'Request Jumlah Server' form. The form has a title bar with 'Home', 'Server', and 'Request' tabs. Below the title bar is the heading 'REQUEST JUMLAH SERVER'. There is a text input field labeled 'Tambah Jumlah Server' with the value '1' and a 'Submit' button. Below this is a section titled 'DATA SERVER - SMA INTI SURABAYA' containing a table with columns 'ID Server', 'Serial Number', and 'State'. The first row of the table has 'INTSUL-3y49Z' in the 'ID Server' column, 'A7B7-019F-3A03-8DEC' in the 'Serial Number' column, and '1' in the 'State' column. The second row has 'INTSUL-Ev0W' in the 'ID Server' column and '0' in the 'State' column. The 'ID Server' column header and the first row's 'ID Server' value are highlighted with red boxes.

Id server tersebut akan digunakan untuk melakukan login pada halaman proktor yang ada pada VHD.

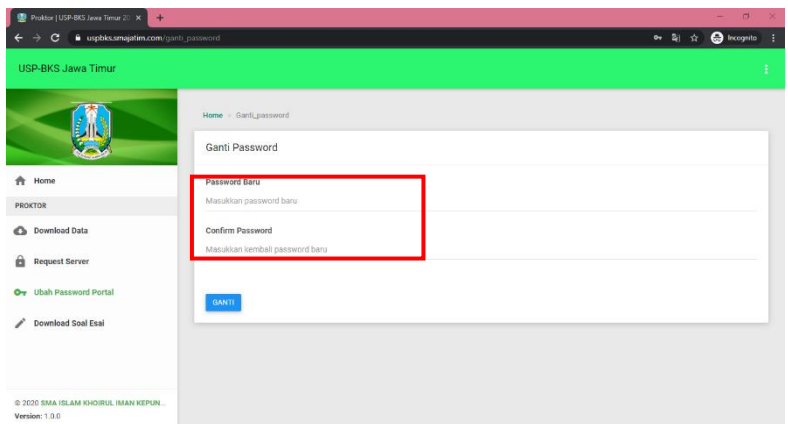
5.4. Menu Ubah Password Portal

Menu ubah password portal digunakan oleh proktor untuk mengubah password agar proktor lebih mudah dalam mengingat password dan lebih aman.

1. Setelah proktor melakukan login, pada halaman utama terdapat menu untuk melakukan ubah password portal.

 **Ubah Password Portal**

2. Klik menu ubah password portal



The screenshot shows a web browser window with the URL usp.bks.unsugfm.com/ganti_password. The page has a green header with the text "USP-BKS Jawa Timur". On the left, there is a sidebar menu with the following items: Home, PROKTOR, Download Data, Request Server, Ubah Password Portal (highlighted with a green key icon), and Download Soal Esai. The main content area is titled "Ganti Password" and contains a form with two input fields: "Password Baru" (Masukkan password baru) and "Confirm Password" (Masukkan kembali password baru). A red rectangle highlights these two input fields. Below the form is a blue button labeled "GANTI". At the bottom left, there is a footer with the text "© 2020 SMA ISLAM KHORUS, IRAN KEPUN. Version: 1.0.0".

3. Masukkan password baru pada kolom yang sudah disediakan. Klik Ganti

Ganti Password

Password Baru
Masukkan password baru

Confirm Password
Masukkan kembali password baru

GANTI

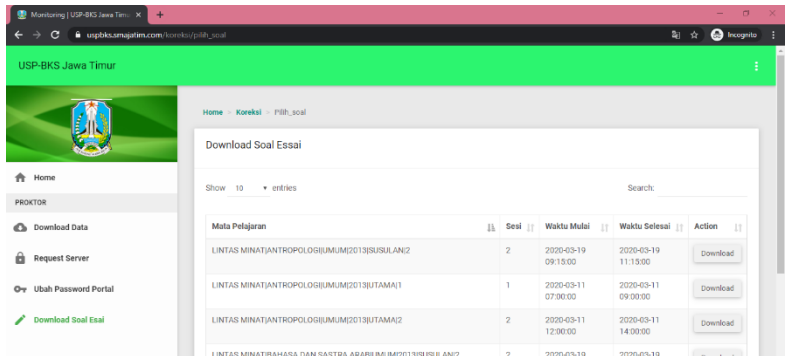
5.5. Menu Download Soal Esai

Menu download soal esai digunakan untuk mengunduh soal esai yang digunakan pada ujian di setiap mata pelajaran.

1. Setelah proktor melakukan login, pada halaman utama terdapat menu untuk mengunduh soal esai.



2. Klik menu download soal esai



- Klik tombol download pada kolom “Action” sesuai dengan mata pelajaran yang diinginkan.

Mata Pelajaran	Sesi	Waktu Mulai	Waktu Selesai	Action
LINTAS MINAT ANTROPOLOGI UMUM 2013 SUSULAN2	2	2020-03-19 09:15:00	2020-03-19 11:15:00	Download
LINTAS MINAT ANTROPOLOGI UMUM 2013 UTAMA1	1	2020-03-11 07:00:00	2020-03-11 09:00:00	Download
LINTAS MINAT ANTROPOLOGI UMUM 2013 UTAMA2	2	2020-03-11 12:00:00	2020-03-11 14:00:00	Download

Berikut contoh dari soal esai yang di download

Soal Essai

K13_Antropologi_2

Kode Soal: 27D

Secara garis besar, kelompok-etnik yang mendiami wilayah Indonesia terbagi dalam enam kelompok besar diantaranya adalah Etnik Jawa. Etnik Jawa ada berada di Pulau Jawa diantaranya ada suku Badui, suku Sunda, suku Jawa, suku Tengger, suku Osing dan suku Betawi. Sekalipun dalam satu pulau, Suku Sunda, Suku Jawa dan suku lainnya terdapat beberapa perbedaan diantaranya adalah penggunaan bahasa. Ada beberapa kata yang sama digunakan baik di Suku Sunda maupun Jawa namun keduanya memiliki arti yang berbeda. Seperti penggunaan kata “gendang” dalam Bahasa Jawa berarti gasing sedang dalam Bahasa Sunda berarti “papa”. Sehingga penggunaan bahasa ini dapat menimbulkan salah sangka dalam komunikasi. Berdasarkan ilustrasi tersebut, berikan dua contoh sikap dalam menghadapi perbedaan tersebut!

Kode Soal: 28D

Cermati intiari hasil penelitian etnografi!

Penelitian ini mengenai konstruksi identitas dan politik identitas etnis Bawean. Intiari etnis Bawean berarti penduduk asli yang mendiami Pulau Bawean, Kabupaten Gresik, Jawa Timur. Tujuan penelitian ini adalah untuk mendeskripsikan konstruksi identitas etnis Bawean dan untuk mendeskripsikan politik identitas etnis Bawean. Hasil penelitian ini menunjukkan bahwa munculnya

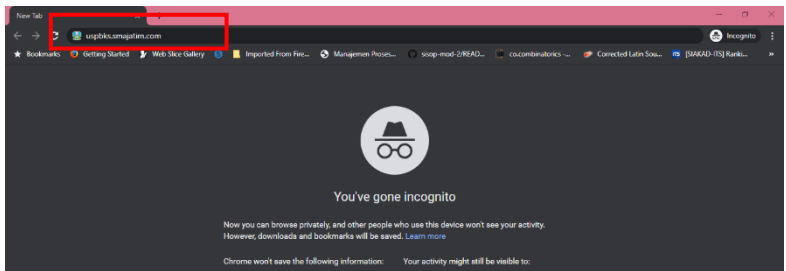
BAB 6

USER GUIDE ADMINISTRATOR

6.1. Halaman Utama Portal Proktor

Administrator merupakan pengguna aplikasi USP-BKS yang memiliki hak akses penuh terhadap aplikasi portal manajemen USP-BKS. Administrator dapat melakukan monitoring dan juga manajemen terhadap semua menu yang ada pada USP-BKS.

1. Administrator mengakses alamat USP-BKS yaitu `uspbks.smajatim.com` pada browser di komputer

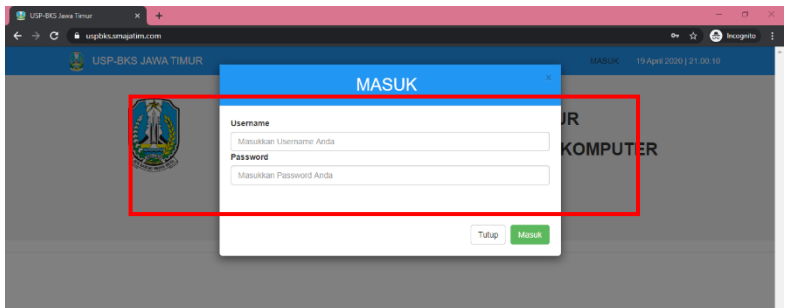


2. Maka akan muncul halaman awal dari USP-BKS

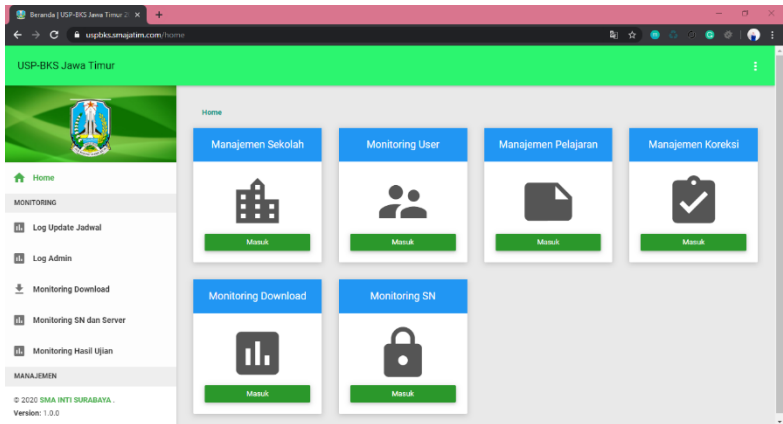


3. Administrator melakukan login dengan memilih menu

MASUK



4. Setelah login, akan muncul halaman utama administrator seperti berikut.



5. Pada halaman administrator menu dibagi menjadi 2 menu utama yaitu menu untuk melakukan monitoring dan menu untuk melakukan manajemen.

- Menu Monitoring

Pada menu ini dibagi lagi menjadi beberapa submenu, yaitu Log Update Jadwal, Log Admin, Monitoring Download, Monitoring SN dan Server, dan Monitoring Hasil Ujian.

- i. Log Update Jadwal, digunakan untuk melakukan monitoring terhadap update jadwal yang pernah dilakukan.
- ii. Log Admin, digunakan untuk melakukan monitoring terhadap perubahan yang dilakukan oleh administrator.

- iii. Monitoring Download, digunakan untuk melakukan monitoring terhadap kegiatan download yang dilakukan oleh pengguna USP-BKS.
- iv. Monitoring SN dan Server, digunakan untuk melakukan monitoring terhadap perubahan yang terjadi pada SN dan Server di setiap sekolah.
- v. Monitoring Hasil Ujian, digunakan untuk melakukan monitoring terhadap hasil ujian setiap siswa di setiap sekolah pada semua mata pelajaran.

- Menu Manajemen

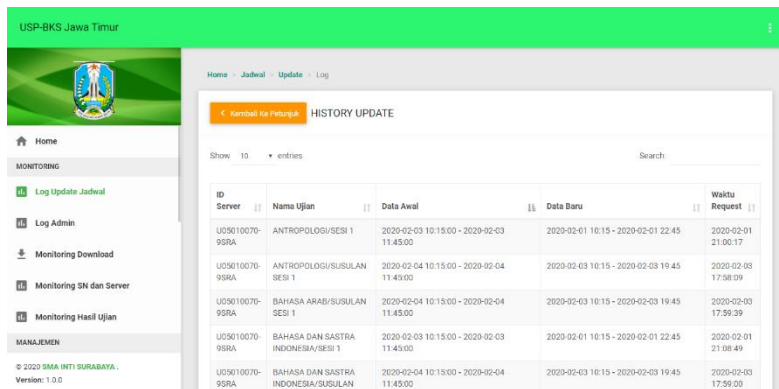
Pada menu ini dibagi lagi menjadi beberapa submenu, yaitu Manajemen SN dan Server, Akses Download Kabupaten/Kota, Manajemen Ujian, Manajemen Paket-Soal, Manajemen Sekolah, Manajemen User, Manajemen Pelajaran, dan Manajemen Siswa.

- i. Manajemen SN dan Server, digunakan untuk melihat, merubah, dan menghapus server yang ada pada setiap sekolah.
- ii. Akses Download Kabupaten / Kota, digunakan untuk merubah izin untuk melakukan download pada kabupaten/kota.
- iii. Manajemen Ujian, digunakan untuk melakukan manajemen terhadap semua mata pelajaran yang akan diujikan.
- iv. Manajemen Paket-Soal, digunakan untuk melakukan manajemen terhadap semua paket soal yang akan diujikan.
- v. Manajemen Sekolah, digunakan untuk melakukan manajemen terhadap semua sekolah yang akan mengikuti ujian.
- vi. Manajemen User, digunakan untuk melakukan manajemen terhadap semua pengguna yang memiliki hak akses terhadap aplikasi USP-BKS.
- vii. Manajemen Pelajaran, digunakan untuk melakukan manajemen terhadap semua pelajaran yang akan diujikan.

viii. Manajemen Siswa, digunakan untuk melakukan manajemen terhadap semua siswa yang akan mengikuti ujian.

6.2. Menu Log Update Jadwal

Menu Log Update Jadwal merupakan menu yang berisi riwayat permintaan izin untuk merubah jadwal yang dilakukan oleh setiap sekolah pada beberapa mata pelajaran yang diinginkan. Data yang ditampilkan antara lain id server, nama ujian, data awal, data baru, dan waktu request.

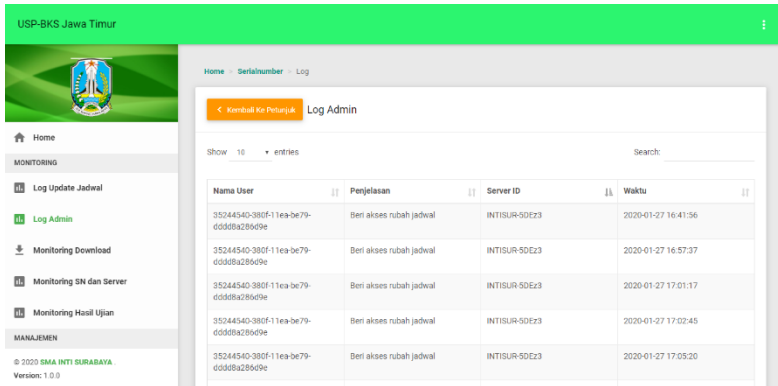


ID Server	Nama Ujian	Data Awal	Data Baru	Waktu Request
U05010070-99RA	ANTROPOLOGI/SESI 1	2020-02-03 10:15:00 - 2020-02-03 11:45:00	2020-02-01 10:15 - 2020-02-01 22:45	2020-02-01 21:00:17
U05010070-99RA	ANTROPOLOGI/SUSULAN SESE 1	2020-02-04 10:15:00 - 2020-02-04 11:45:00	2020-02-03 10:15 - 2020-02-03 19:45	2020-02-03 17:58:09
U05010070-99RA	BAHASA ARAB/SUSULAN SESE 1	2020-02-04 10:15:00 - 2020-02-04 11:45:00	2020-02-03 10:15 - 2020-02-03 19:45	2020-02-03 17:59:39
U05010070-99RA	BAHASA DAN SASTRA INDONESIA/SESE 1	2020-02-03 10:15:00 - 2020-02-03 11:45:00	2020-02-01 10:15 - 2020-02-01 22:45	2020-02-01 21:08:49
U05010070-99RA	BAHASA DAN SASTRA INDONESIA/SUSULAN	2020-02-04 10:15:00 - 2020-02-04 11:45:00	2020-02-03 10:15 - 2020-02-03 19:45	2020-02-03 17:59:00

6.3. Menu Log Admin

Menu Log Admin merupakan menu yang berisi riwayat kegiatan yang dilakukan oleh administrator, mulai

dari riwayat pemberian akses untuk merubah jadwal, menghapus server, mereset SN, dan lain sebagainya. Data yang ditampilkan pada menu ini antara lain adalah nama user, penjelasan, server id, dan waktu.



USP-BKS Jawa Timur

Home Serialnumber Log

Log Admin

Show 10 entries Search:


Nama User	Penjelasan	Server ID	Waktu
35244540-380f-11ea-be79-65558a289d9e	Beri akses rubah jadwal	INTISUR-SDE23	2020-01-27 16:41:56
35244540-380f-11ea-be79-65558a289d9e	Beri akses rubah jadwal	INTISUR-SDE23	2020-01-27 16:57:37
35244540-380f-11ea-be79-65558a289d9e	Beri akses rubah jadwal	INTISUR-SDE23	2020-01-27 17:01:17
35244540-380f-11ea-be79-65558a289d9e	Beri akses rubah jadwal	INTISUR-SDE23	2020-01-27 17:02:45
35244540-380f-11ea-be79-65558a289d9e	Beri akses rubah jadwal	INTISUR-SDE23	2020-01-27 17:05:20

© 2020 SMA INTI SURABAYA
Version: 1.0.0

6.4. Menu Monitoring Download

Menu Monitoring Download merupakan menu yang berisi riwayat download yang dilakukan oleh setiap sekolah. Data yang ditampilkan pada menu ini antara lain adalah nama sekolah, username, tipe, nama file, dan juga tanggal download.

USP-BKS Jawa Timur



Monitoring Download

Monitoring SN dan Server

Monitoring Hasil Ujian

MANAJEMEN

Manajemen SN dan Server

Akses Download Kabupaten / Kota

Manajemen Ujian

Manajemen Paket Soal

© 2020 SMA INTI SURABAYA

Version: 1.0.0

Home Download Monitor

Show 10 entries


Search

Sekolah	Username	Tipe	Nama File	Tanggal
SMA NEGERI 1 CLURING	U05340033	master	master_all	29 Feb 2020 16:05
SMA NEGERI 2 PASURUAN	U05070005	master	master_all	29 Feb 2020 17:06
SMA NEGERI 1 GRESIK	U05100001	master	master_all	29 Feb 2020 17:58
SMA NEGERI 2 NGAWI	U05180008	master	master_all	29 Feb 2020 18:47
SMA MUHAMMADRIYAH 3 JEMBER	U05330007	master	master_all	29 Feb 2020 19:46
SMA NAZIHATUT THULAB	U05360007	master	master_all	29 Feb 2020 21:52
SMA NEGERI 1 PACE	U05230007	master	master_all	01 Mar 2020 15:31
SMA NEGERI 1 PILANGKENCENG	U05170003	master	master_all	11 Mar 2020 12:34

6.5. Menu Monitoring SN dan Server

Menu Monitoring SN dan Server merupakan menu yang berisi data sekolah beserta id server dan serial numbernya. Pada menu ini juga menyimpan jumlah dari izin merubah jadwal dan jumlah reset SN yang pernah dilakukan oleh setiap sekolah. Pada menu ini juga dapat memberikan aksi untuk melakukan reset SN, menghapus Id Server, dan memberikan izin untuk merubah jadwal.

USP-BKS Jawa Timur



Monitoring SN dan Server

Monitoring Hasil Ujian

MANAJEMEN

Manajemen SN dan Server

Akses Download Kabupaten / Kota

Manajemen Ujian

Manajemen Paket-Soal

Manajemen Sekolah

© 2020 SMA INTI SURABAYA

Version: 1.0.0

Home > Beranda

Monitoring Server

Show 10 entries

Search:

ID Server	Nama Sekolah	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010000-05NI	SMA INTI SURABAYA	E915-30F1-CF07-ECBC	0	0	<div>Reset SN</div> <div>Delete ID</div> <div>Izinkan Ubah Jadwal</div>
U05010000-1RDW	SMA INTI SURABAYA	B84A-0317-108D-08F7	1	0	<div>Reset SN</div> <div>Delete ID</div> <div>Cabut Izin</div>
U05010000-252M	SMA INTI SURABAYA	B8F2-5554-81BE-799D	0	0	<div>Reset SN</div> <div>...</div>

Untuk melakukan reset SN, tekan tombol “Reset SN” yang ada pada kolom “Action” sesuai dengan sekolah yang diinginkan.

ID Server	Nama Sekolah	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010000-05NI	SMA INTI SURABAYA	E915-30F1-CF07-ECBC	0	0	<div>Reset SN</div> <div>Delete ID</div> <div>Izinkan Ubah Jadwal</div>

Untuk menghapus Id Server, tekan tombol “Delete ID” yang ada pada kolom “Action” sesuai dengan sekolah yang diinginkan.

ID Server	Nama Sekolah	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010000-05NI	SMA INTI SURABAYA	E915-30F1-CF07-ECBC	0	0	<div>Reset SN</div> <div>Delete ID</div> <div>Izinkan Ubah Jadwal</div>

46

Untuk memberikan izin merubah jadwal, tekan tombol “Izinkan Ubah Jadwal” yang ada pada kolom “Action” sesuai dengan sekolah yang diinginkan.

ID Server	Nama Sekolah	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010000-05NI	SMA INTI SURABAYA	E915-30F1-CF07-ECBC	0	0	Reset SN Delete ID Izinkan Ubah Jadwal

6.6. Menu Monitoring Hasil Ujian

Menu Monitoring Hasil Ujian merupakan menu yang berisi hasil dari ujian semua siswa pada semua mata pelajaran di semua SMA Negeri yang ada pada kabupaten/kota di Provinsi Jawa Timur. Pada halaman awal pada menu ini, akan ditampilkan semua daftar kabupaten/kota yang ada pada Provinsi Jawa Timur.

USP-BKS Jawa Timur

Home Kota Hasil

Show 10 entries

Kode Rayon	Kota / Kabupaten	Aksi
01	KOTA SURABAYA	Lihat
02	KOTA MALANG	Lihat
03	KOTA MADIUN	Lihat
04	KOTA KEDIRI	Lihat
05	KOTA MOJOKERTO	Lihat
06	KOTA BLITAR	Lihat
07	KOTA PASURUAN	Lihat

© 2020 SMA INTI SURABAYA
Version: 1.0.0

Kemudian klik “Lihat” pada kabupaten / kota yang dikehendaki.

Kode Rayon	Kota / Kabupaten	Aksi
01	KOTA SURABAYA	Lihat

Akan ditampilkan semua SMA Negeri yang ada pada kabupaten / kota yang telah dipilih.

USP-BKS Jawa Timur

Home Hasil Ujian 01

HASIL KOTA SURABAYA

Show 10 entries

No	Nama Sekolah	Aksi
1	SMA KHADIJAH	Lihat
2	SMA KARTIKA WUJAYA SURABAYA	Lihat
3	SMA KRISTEN PETRA 5	Lihat
4	SMA SANTO CAROLUS	Lihat
5	SMA KRISTEN PETRA 1 SURABAYA	Lihat
6	SMA KRISTEN GLORIA 1	Lihat
7	SMA CIPUTRA	Lihat

© 2020 SMA INTI SURABAYA. Version: 1.0.0

Kemudian klik “Lihat” lagi pada SMA yang dikehendaki untuk melihat daftar siswa yang ada pada SMA tersebut beserta daftar mata pelajaran dan hasil ujian dari siswa yang bersangkutan.

No	Nama Sekolah	Aksi
1	SMA KHADIJAH	Lihat

USP-BKS Jawa Timur

Home Hasil All 00716028-3c7c-11ea-b872-dc7195bc523b

HASIL UJIAN SISWA SMA KHADIJAH

Pilih Mata Pelajaran

Show 10 entries Search

No	Nama Siswa	Username	Mata Pelajaran	Jumlah Benar	Jumlah Salah	Jumlah Kosong	Nilai	Created At
1	ZAMROTUS SHOHFIH	U050101340432	UMUMIPENDIDIKAN AGAMA ISLAM DAN BUDI PEKERTI(UMJUM2013)UTAMA1	33	7	0	82.5	2020-02-27 22:30:35
2	WIYANGGA KHULIF KATANHIA	U050101340430	UMUMIPENDIDIKAN AGAMA ISLAM DAN BUDI PEKERTI(UMJUM2013)UTAMA1	35	5	0	87.5	2020-02-27 22:30:35
3	ZAKIYAH SHABIRINA CAHYANI	U050101340431	UMUMIPENDIDIKAN AGAMA ISLAM DAN BUDI PEKERTI(UMJUM2013)UTAMA1	32	8	0	80	2020-02-27 22:30:35
4	TENGGU KATULAMPATI	U050101340429	UMUMIPENDIDIKAN AGAMA ISLAM DAN BUDI PEKERTI(UMJUM2013)UTAMA1	36	4	0	90	2020-02-27 22:30:35

© 2020 SMA INTI SURABAYA
Version: 1.0.0

6.7. Menu Manajemen SN dan Server

Menu Manajemen SN dan Server merupakan menu yang digunakan untuk mengatur server yang ada pada setiap sekolah. Pada menu ini dapat melihat, mengubah, dan juga menghapus server yang ada pada sekolah.

USP-BKS Jawa Timur

Home Serialnumber Sekolah

Manajemen SN dan Server

Akses Download Kabupaten / Kota

Manajemen Ujian

Manajemen Paket-Soal

Manajemen Sekolah

Manajemen User

Manajemen Pelajaran

© 2020 SMA INTI SURABAYA
Version: 1.0.0

Show 10 entries Search

Nomor	ID	Nama Sekolah	Alamat	Jumlah Server	Action
1	U05010134	SMA KHADIJAH	JL. A. YANI 2-4 SURABAYA	4	Edit Hapus
2	U05010135	SMA KARTIKA WILAJA SURABAYA	JL. KARAH 182 SURABAYA	1	Edit Hapus
3	U05010136	SMA KRISTEN PETRA 5	JL. JEMUR ANDAYANI XVII/2	3	Edit Hapus

Untuk melihat data lengkap server yang ada pada sekolah tersebut, klik pada tombol “Lihat”.

Nomor	ID	Nama Sekolah	Alamat	Jumlah Server	Action
1	U05010134	SMA KHADIJAH	JL. A. YANI 2-4 SURABAYA	4	<div>Lihat</div> <div>Edit</div> <div>Hapus</div>

Pada data server yang ditampilkan, dapat dilakukan beberapa aksi yaitu memberikan izin untuk merubah jadwal, reset SN, dan delete ID.

USP-BKS Jawa Timur

Home Serialnumber Show_sn 00016528-3e7e-11ea-bf72-8c7198b0523b

DATA SERVER - SMA KHADIJAH

ID Server	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010134-0YOF	5C96-B237-4600-B260	Izin tidak diberikan	0	<div>Izinkan Ubah Jadwal</div> <div>Reset SN</div> <div>Delete ID</div>
U05010134-TUSQ	845E-5D67-4355-D1E1	Izin tidak diberikan	0	<div>Izinkan Ubah Jadwal</div> <div>Reset SN</div> <div>Delete ID</div>
U05010134-2LC6	ACFB-CF44-664D-B523	Izin tidak diberikan	0	<div>Izinkan Ubah Jadwal</div> <div>Reset SN</div> <div>Delete ID</div>
U05010134-3D6Q	2AEE-EEED-D1C6-C1CF	Izin tidak diberikan	0	<div>Izinkan Ubah Jadwal</div> <div>Reset SN</div> <div>Delete ID</div>

© 2020 SMA INTI SURABAYA. Version: 1.0.0

Untuk memberikan izin ubah jadwal pada sebuah server, klik tombol “Izinkan Ubah Jadwal” pada server yang dikehendaki.

ID Server	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010134-0YOF	5C96-B237-4600-B260	Izin tidak diberikan	0	<div>Izinkan Ubah Jadwal</div> <div>Reset SN</div> <div>Delete ID</div>

Untuk melakukan reset SN pada server sekolah, klik tombol “Reset SN” pada kolom “Action” pada server yang dikehendaki.

ID Server	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010134-0YOF	5C96-B237-4600-B260	Izin tidak diberikan	0	<input type="button" value="Izinkan Ubah Jadwal"/> <input type="button" value="Reset SN"/> <input type="button" value="Delete ID"/>

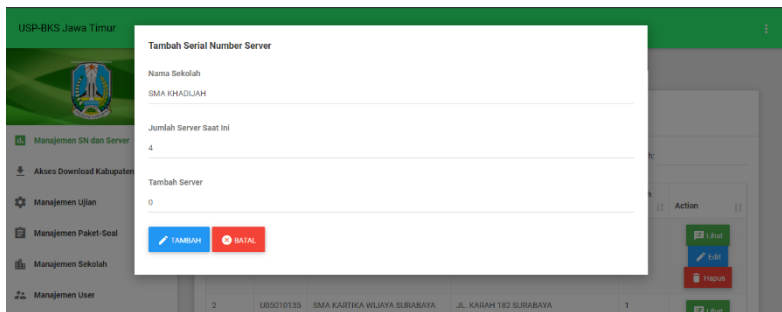
Sedangkan untuk menghapus Id server, klik tombol “Delete ID” pada kolom “Action” pada server yang dikehendaki.

ID Server	Serial Number	Izin merubah jadwal	Jumlah Reset SN	Action
U05010134-0YOF	5C96-B237-4600-B260	Izin tidak diberikan	0	<input type="button" value="Izinkan Ubah Jadwal"/> <input type="button" value="Reset SN"/> <input type="button" value="Delete ID"/>

Sedangkan untuk menambah jumlah server yang ada pada sekolah tersebut, klik pada tombol “Edit”.

Nomor	ID	Nama Sekolah	Alamat	Jumlah Server	Action
1	U05010134	SMA KHADIJAH	JL. A. YANI 2-4 SURABAYA	4	<input type="button" value="Lihat"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Isi data yang di perlukan pada form yang disediakan, kemudian klik “Tambah”



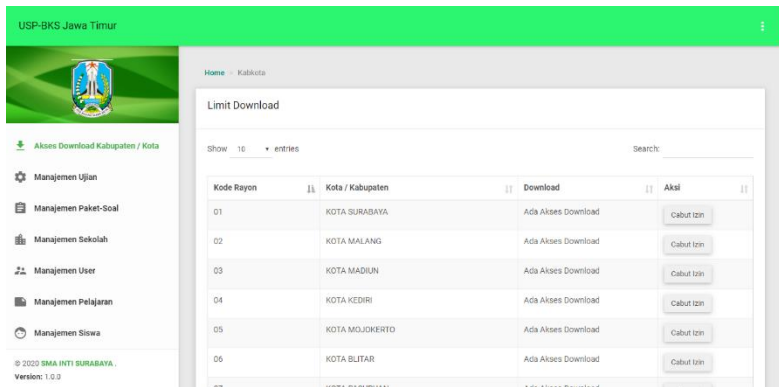
Sedangkan untuk menghapus server yang ada pada sekolah tersebut, klik pada tombol “Hapus”.

Nomor	ID	Nama Sekolah	Alamat	Jumlah Server	Action
1	U05010134	SMA KHADIJAH	JL. A. YANI 2-4 SURABAYA	4	<div> <div>Lihat</div> <div>Edit</div> <div>Hapus</div> </div>

6.8. Menu Akses Download Kabupaten/Kota

Menu Akses Download Kabupaten/Kota merupakan menu yang digunakan untuk memberikan atau mencabut izin

download pada kabupaten/kota yang dikehendaki.

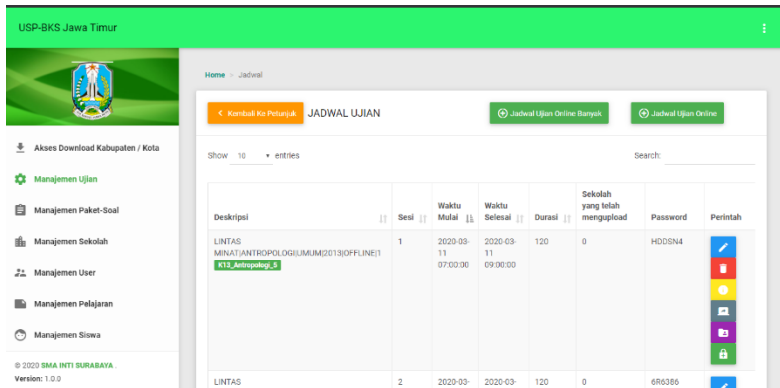


Untuk mencabut izin, klik tombol “Cabut Izin” pada kolom “Aksi” yang ada pada kabupaten/kota yang dikehendaki.

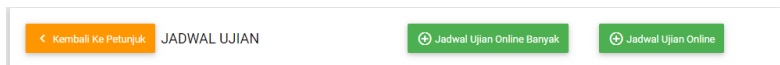
Kode Rayon	Kota / Kabupaten	Download	Aksi
01	KOTA SURABAYA	Ada Akses Download	Cabut Izin

6.9. Menu Manajemen Ujian

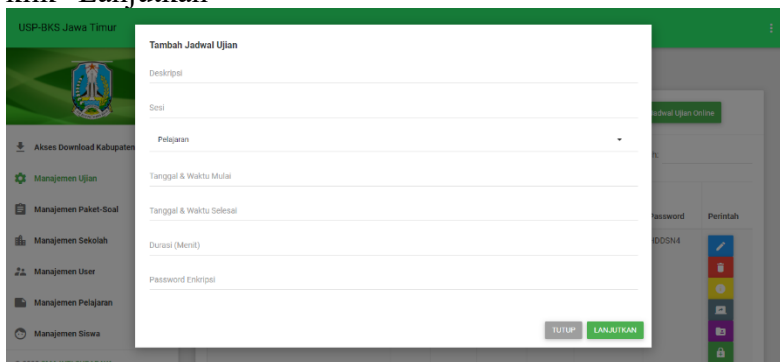
Menu Manajemen Ujian merupakan menu yang digunakan untuk mengatur ujian yang akan dilakukan. Pada menu ini dapat menambahkan, mengubah, dan menghapus jadwal ujian.



Untuk menambahkan jadwal ujian, klik “Jadwal Ujian Online”



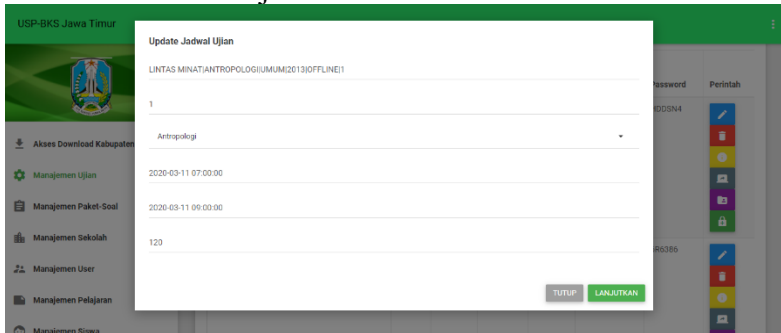
Isi data yang diperlukan pada form yang muncul. Kemudian klik “Lanjutkan”



Sedangkan untuk mengubah jadwal yang sudah ada, klik tombol biru

Deskripsi	Sesi	Waktu Mulai	Waktu Selesai	Durasi	Sekolah yang telah mengupload	Password	Perintah
LINTAS MINAT ANTROPOLOGI UMUM 2013 OFFLINE 1 K13_Antropologi_5	1	2020-03-11 07:00:00	2020-03-11 09:00:00	120	0	HDDSN4	     

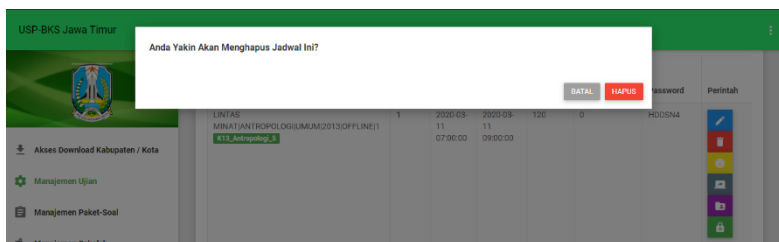
Isi data yang diperlukan pada form yang disediakan, kemudian klik “Lanjutkan”



Sedangkan untuk menghapus jadwal yang sudah ada, klik tombol merah

Deskripsi	Sesi	Waktu Mulai	Waktu Selesai	Durasi	Sekolah yang telah mengupload	Password	Perintah
LINTAS MINAT ANTROPOLOGI UMUM 2013 OFFLINE 1 K13_Antropologi_5	1	2020-03-11 07:00:00	2020-03-11 09:00:00	120	0	HDDSN4	     

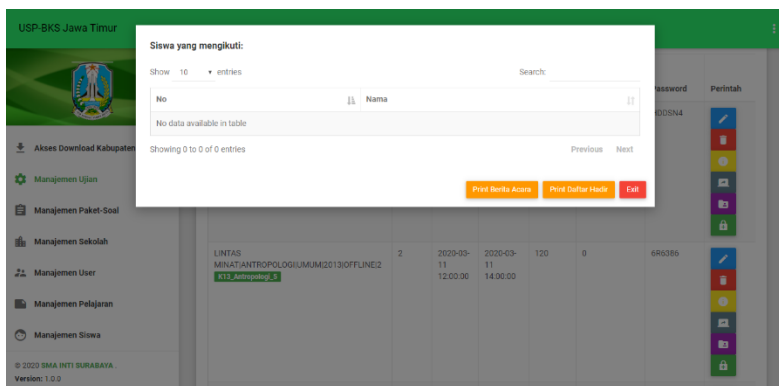
Jika sudah yakin ingin menghapus jadwal tersebut, klik tombol “Hapus”



Sedangkan untuk melihat peserta ujian tersebut, klik tombol kuning.

Deskripsi	Sesi	Waktu Mulai	Waktu Selesai	Durasi	Sekolah yang telah mengupload	Password	Perintah
LINTAS MINAT[ANTROPOLOGI UMUM 2013 OFFLINE]1 K13_Antropologi_5	1	2020-03-11 07:00:00	2020-03-11 09:00:00	120	0	HDDSN4	     

Kemudian akan muncul modal sebagai berikut. Pada modal ini juga terdapat fitur untuk mencetak berita acara dan daftar hadir.



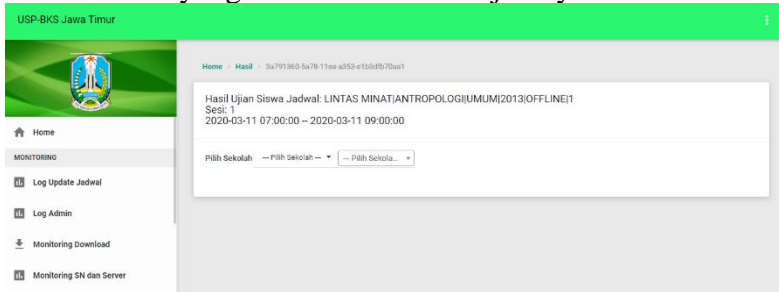
Sedangkan untuk melakukan monitoring terhadap ujian yang sedang berlangsung, klik tombol abu-abu.

Deskripsi	Sesi	Waktu Mulai	Waktu Selesai	Durasi	Sekolah yang telah mengupload	Password	Perintah
LINTAS MINAT[ANTROPOLOGI UMUM 2013 OFFLINE]1 K13_Antropologi_5	1	2020-03-11 07:00:00	2020-03-11 09:00:00	120	0	HDDSN4	

Kemudian untuk melihat hasil ujian, klik tombol berwarna ungu.

Deskripsi	Sesi	Waktu Mulai	Waktu Selesai	Durasi	Sekolah yang telah mengupload	Password	Perintah
LINTAS MINAT[ANTROPOLOGI UMUM 2013 OFFLINE]1 K13_Antropologi_5	1	2020-03-11 07:00:00	2020-03-11 09:00:00	120	0	HDDSN4	







Pilih sekolah yang akan dilihat hasil ujiannya



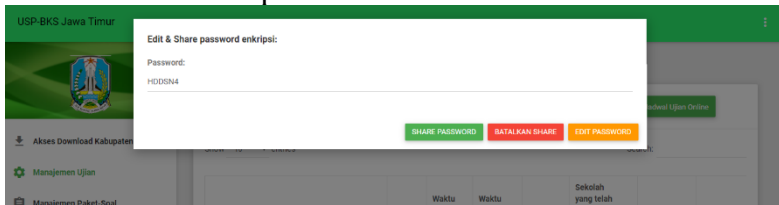
Maka akan muncul hasil ujian pada sekolah yang sudah dipilih.



Sedangkan untuk mengubah kunci untuk membuka ujian, klik tombol berwarna hijau.

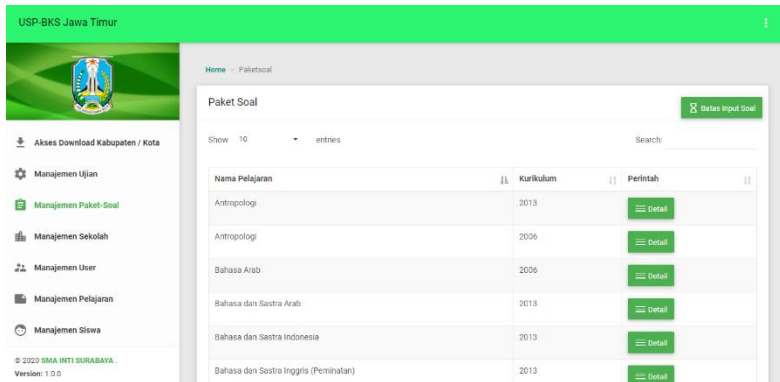
Deskripsi	Sesi	Waktu Mulai	Waktu Selesai	Durasi	Sekolah yang telah mengupload	Password	Perintah
LINTAS MINAT ANTROPOLOGI UMUM 2013 OFFLINE 1 K13_Antropologi_5	1	2020-03-11 07:00:00	2020-03-11 09:00:00	120	0	HDDSN4	     

Pada submenu ini selain digunakan untuk mengubah password, juga dapat melakukan share password, dan membatalkan share password.



6.10. Menu Manajemen Paket-Soal

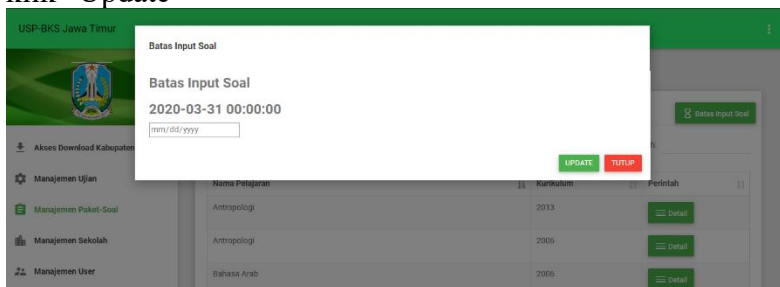
Menu Manajemen Paket-Soal merupakan menu yang digunakan untuk mengatur paket soal yang akan digunakan pada saat ujian berlangsung. Pada menu ini terdapat fitur untuk mengatur batas input soal, melihat detail paket soal, mengubah maupun menghapus paket soal tersebut.



Untuk merubah batas input soal, klik tombol “Batas Input Soal” yang ada pada pojok kanan atas.



Ubah batas input soal dengan tanggal yang sesuai, kemudian klik “Update”



Sedangkan untuk melihat detail dari paket soal, klik tombol “Detail” pada mata pelajaran yang dikehendaki.

Nama Pelajaran	Kurikulum	Perintah
Antropologi	2013	Detail

Kemudian akan muncul daftar paket dari mata pelajaran yang dipilih. Pada halaman ini dapat melakukan penambahan paket, mengubah, maupun menghapus paket soal pada mata pelajaran tersebut.

USP-BKS Jawa Timur

Home / Paketsoal / 34

Pelajaran : Semua [Tambah Paket](#)

Show 10 entities Search:

Nomor	Nama	Pelajaran	Jumlah Soal	
1	K13_Antropologi_1	Antropologi	45	Edit Delete Add Detail Refresh
2	K13_Antropologi_2	Antropologi	45	Edit Delete Add Detail Refresh
3	K13_Antropologi_3	Antropologi	45	Edit Delete Add Detail Refresh
4	K13_Antropologi_4	Antropologi	45	Edit Delete Add Detail Refresh
5	K13_Antropologi_5	Antropologi	45	Edit Delete Add Detail Refresh

© 2020 SMA INTI SURABAYA. Version: 1.0.0

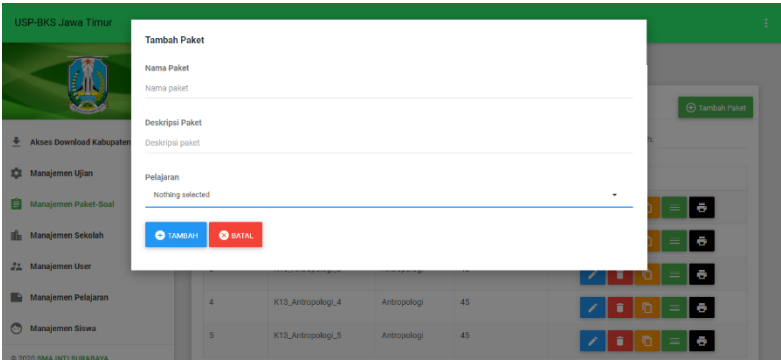
Optimized 1 to 6 of 6 entities

Download [Print](#) [Close](#)

Untuk menambah paket soal, tekan tombol “Tambah Paket” yang ada pada pojok kanan atas.

Pelajaran : Semua [Tambah Paket](#)

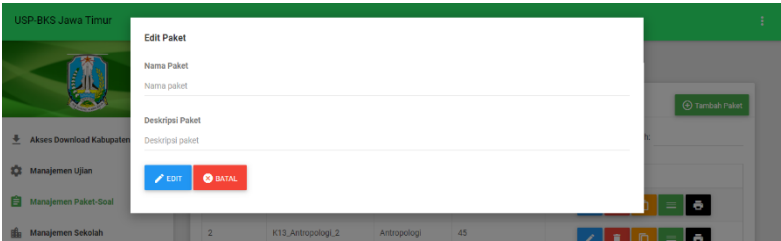
Isi data yang diperlukan pada form yang disediakan, kemudian klik “Tambah”.



Sedangkan untuk mengubah paket soal yang sudah ada, klik tombol biru.

Nomor	Nama	Pelajaran	Jumlah Soal	
1	KT3_Antropologi_1	Antropologi	45	    

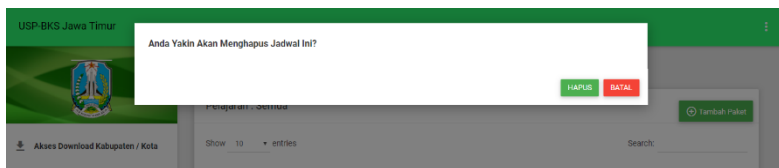
Isi data yang diperlukan pada form yang disediakan, kemudian klik tombol “Edit” untuk menyimpan perubahan yang dilakukan.



Sedangkan untuk menghapus paket soal yang sudah ada, klik tombol berwarna merah.

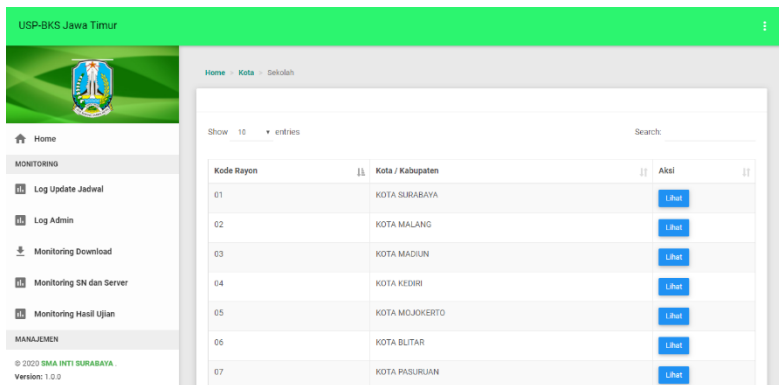
Nomor	Nama	Pelajaran	Jumlah Soal	
1	K13_AntropologiL1	Antropologi	45	    

Jika sudah yakin untuk menghapus paket soal tersebut, klik tombol “Hapus” pada modal konfirmasi yang muncul.



6.11. Menu Manajemen Sekolah

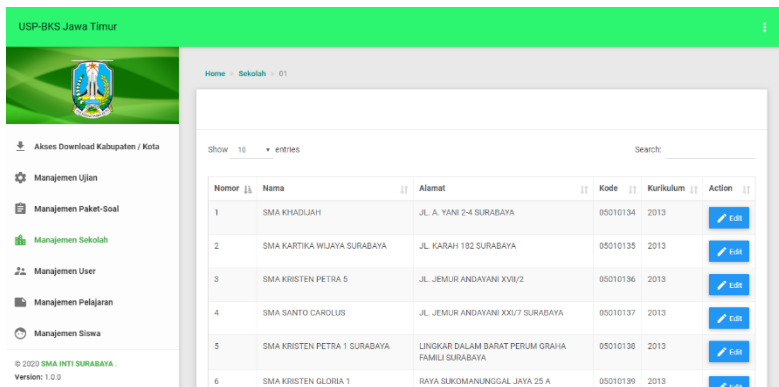
Menu Manajemen Sekolah merupakan menu yang digunakan untuk mengelola data sekolah yang ikut berpartisipasi dalam USP-BKS. Pada menu ini dapat melihat semua data sekolah pada setiap kabupaten/kota dan dapat merubah data sekolah tersebut.



Untuk melihat daftar sekolah pada sebuah kabupaten/kota, tekan tombol “Lihat” pada kolom aksi sesuai dengan kabupaten/kota yang dipilih.

Kode Rayon	Kota / Kabupaten	Aksi
01	KOTA SURABAYA	Lihat

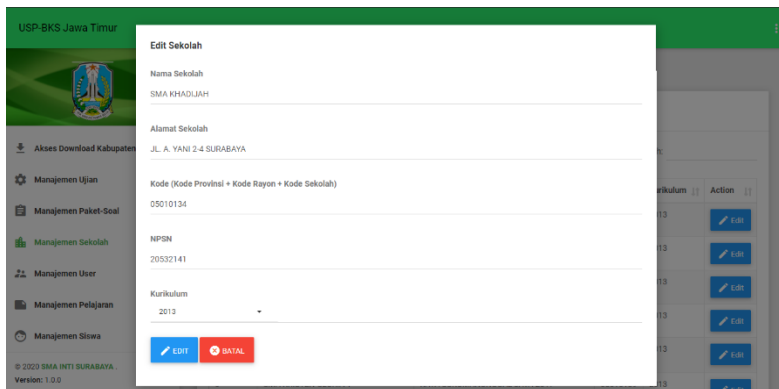
Kemudian akan ditampilkan semua sekolah yang ada pada kabupaten/kota yang dipilih.



Untuk mengubah data sekolah tekan tombol “Edit” pada kolom “Action” sesuai dengan sekolah yang dipilih.

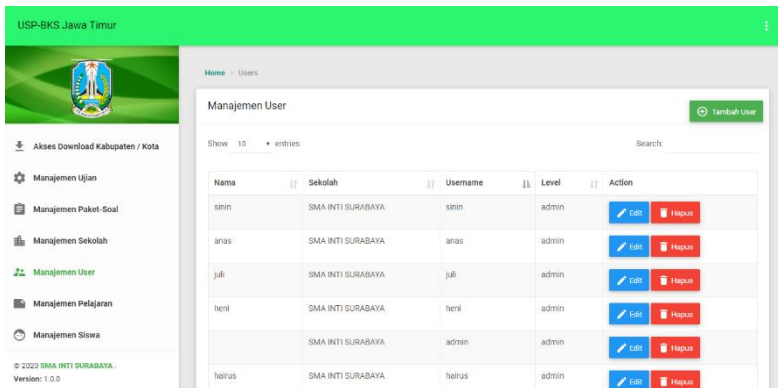
Nomor	Nama	Alamat	Kode	Kurikulum	Action
1	SMA KHADIJAH	JL. A. YANI 2-4 SURABAYA	05010134	2013	Edit

Isi data yang diperlukan pada form yang disediakan. Tekan tombol “Edit” untuk menyimpan perubahan data yang dilakukan.



6.12. Menu Manajemen User

Menu Manajemen User merupakan menu yang digunakan untuk mengelola user yang menggunakan sistem ini. Pada menu ini terdapat fitur untuk menambah, mengubah dan menghapus pengguna.



The screenshot displays the 'Manajemen User' interface. On the left is a sidebar with a green header 'USP-BKS Jawa Timur' and a list of menu items: 'Akses Download Kabupaten / Kota', 'Manajemen Ujian', 'Manajemen Paket- Soal', 'Manajemen Sekolah', 'Manajemen User' (highlighted in green), 'Manajemen Pelajaran', and 'Manajemen Siswa'. The footer of the sidebar indicates '© 2023 SMA INTI SURABAYA' and 'Version: 1.0.0'. The main content area has a green header 'Manajemen User' and a '+ Tambah User' button. Below this is a table with columns: Nama, Sekolah, Username, Level, and Action. The table contains six rows of user data.

Nama	Sekolah	Username	Level	Action
setin	SMA INTI SURABAYA	setin	admin	Edit Hapus
arias	SMA INTI SURABAYA	arias	admin	Edit Hapus
juli	SMA INTI SURABAYA	juli	admin	Edit Hapus
heni	SMA INTI SURABAYA	heni	admin	Edit Hapus
	SMA INTI SURABAYA	admin	admin	Edit Hapus
haiyus	SMA INTI SURABAYA	haiyus	admin	Edit Hapus


Untuk menambahkan pengguna, tekan tombol “Tambah User” pada pojok kanan atas.



This image is a close-up of the top right corner of the 'Manajemen User' page, showing the header 'Manajemen User' and the green '+ Tambah User' button.

Isi data yang diperlukan pada form yang telah disediakan. Jika sudah selesai mengisi, tekan tombol “Submit” untuk menyimpan data yang sudah dimasukkan.

USP-BKS Jawa Timur



Akses Download Kabupaten / Kota

Manajemen Ujian

Manajemen Paket-Soal

Manajemen Sekolah

Manajemen User

Manajemen Pelajaran

Manajemen Siswa

© 2020 SMA INTI SURABAYA.
Version: 1.0.0

Manajemen Pelajaran

Manajemen Siswa

© 2020 SMA INTI SURABAYA.
Version: 1.0.0

Create User

Sekolah:

Level:

Username:

Password:

Nama:

NISN:

Foto: No file chosen

Jenis Kelamin:

Tempat Lahir:

Tempat Lahir:

Tanggal Lahir:

Sedangkan untuk mengubah data pengguna yang sudah ada, tekan tombol “Edit”.

Nama	Sekolah	Username	Level	Action
sinin	SMA INTI SURABAYA	sinin	admin	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Isi data yang diperlukan pada form yang disediakan, jika sudah selesai mengisi tekan tombol “Submit” untuk menyimpan perubahan yang dilakukan.

USP-BKS Jawa Timur

Home Users 17896930-4592-11ea-9587-16d6fbc4542 Edit

Edit User

Sekolah SMA INTI SURABAYA

Level Admin

Username sinin

Password *****

Nama sinin

NISN NISN

Foto [Choose File](#) No file chosen

Jenis Kelamin *

Tempat Lahir Tempat Lahir

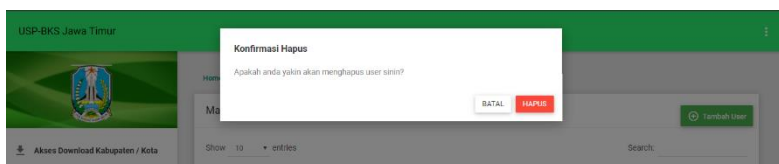
Tanggal Lahir contoh: 30/07/2016

[Simpan](#)

Untuk menghapus data pengguna yang sudah ada, klik tombol “Hapus”

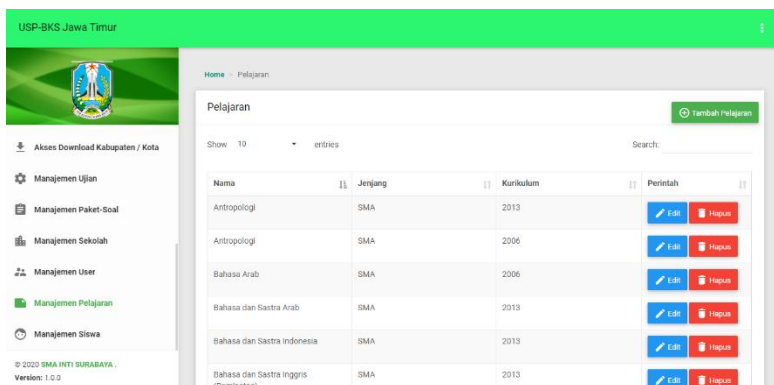
Nama	Sekolah	Username	Level	Action
sinin	SMA INTI SURABAYA	sinin	admin	Edit Hapus

Jika sudah yakin untuk menghapus data tersebut, klik tombol “Hapus” pada modal konfirmasi.



6.13. Menu Manajemen Pelajaran

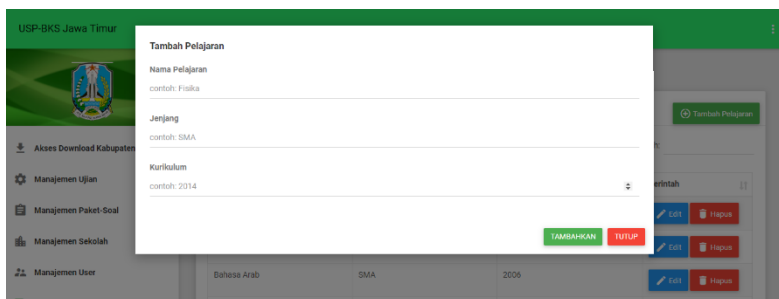
Menu Manajemen Pelajaran merupakan menu yang digunakan untuk mengelola pelajaran yang akan diujikan pada saat ujian berlangsung. Pada menu ini terdapat fitur untuk menambahkan, mengubah dan menghapus pelajaran.



Untuk menambahkan pelajaran, tekan tombol “Tambah Pelajaran” pada pojok kanan atas.



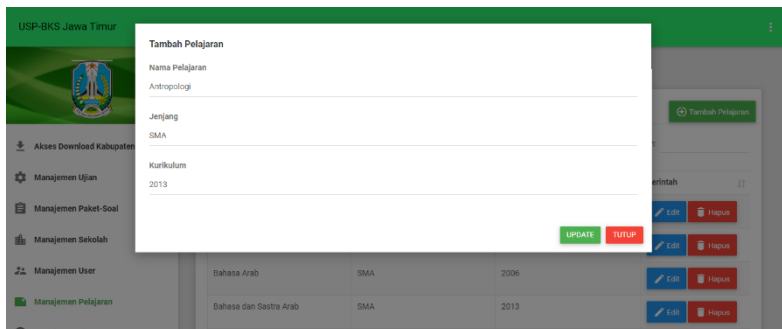
Isi data yang diperlukan pada form yang disediakan. Jika sudah selesai mengisi, klik “Tambahkan” untuk menyimpan data tersebut.



Sedangkan untuk mengubah data pelajaran yang sudah ada, klik tombol “Edit” pada kolom perintah sesuai dengan pelajaran yang ingin diubah.

Nama	Jenjang	Kurikulum	Perintah
Antropologi	SMA	2013	Edit Hapus

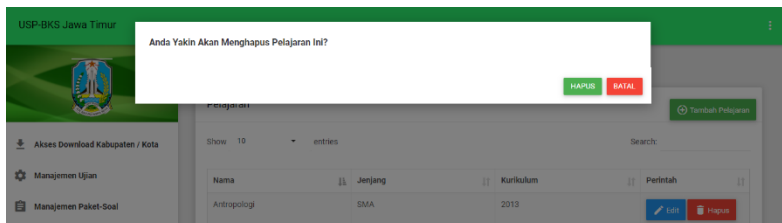
Isi data yang diperlukan pada form yang disediakan. Jika sudah selesai mengisi, klik “Update” untuk menyimpan perubahan yang dilakukan.



Untuk menghapus data pelajaran yang sudah ada, klik tombol “Hapus”

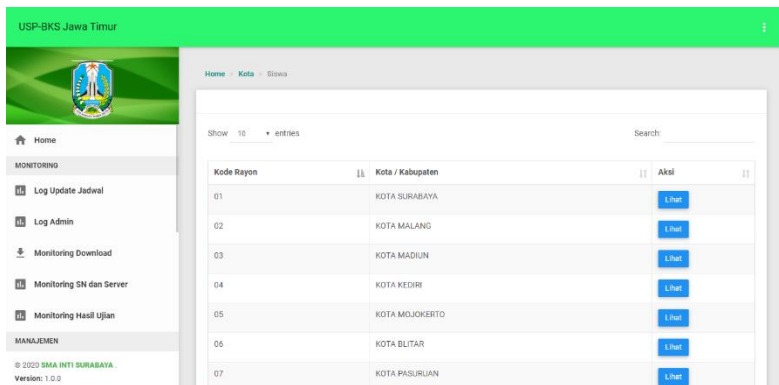
Nama	Jenjang	Kurikulum	Perintah
Antropologi	SMA	2013	Edit Hapus

Jika sudah yakin akan menghapus data tersebut, klik tombol “Hapus” pada modal konfirmasi



6.14. Menu Manajemen Siswa

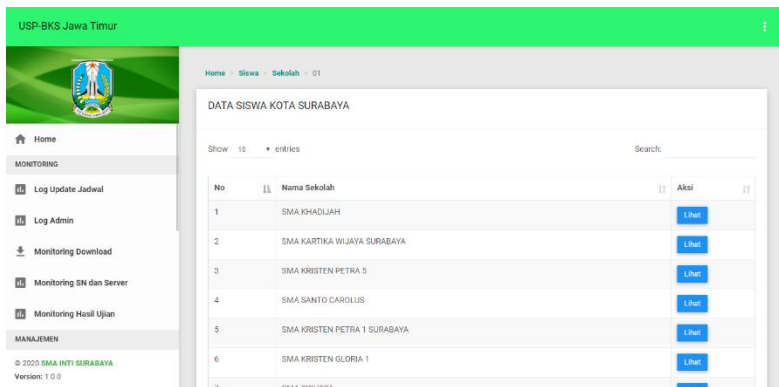
Menu Manajemen Siswa merupakan menu yang digunakan untuk mengelola data siswa yang akan mengikuti ujian. Pada menu ini dapat melihat data siswa pada seluruh SMA Negeri di setiap kabupaten/kota.



Untuk melihat daftar SMA Negeri yang ada pada kota/kabupaten yang diinginkan, tekan tombol “Lihat” pada kolom “Aksi”

Kode Rayon	Kota / Kabupaten	Aksi
01	KOTA SURABAYA	Lihat

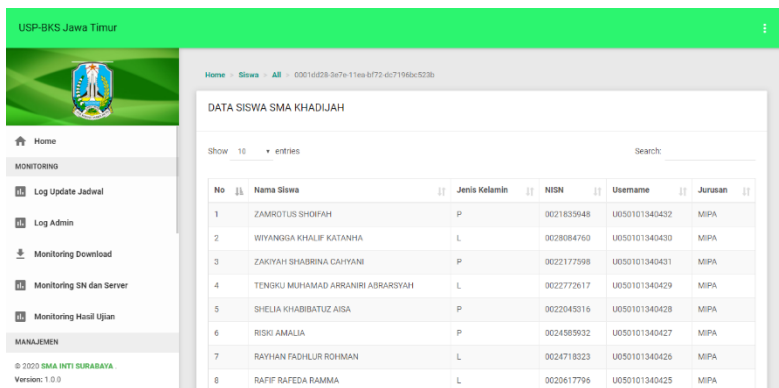
Akan ditampilkan daftar seluruh SMA Negeri yang ada pada kota/kabupaten yang dipilih.



Untuk melihat daftar siswa pada SMA Negeri yang diinginkan, tekan tombol “Lihat” pada kolom “Aksi”

No	Nama Sekolah	Aksi
1	SMA KHADIJAH	Lihat

Dan akan ditampilkan daftar seluruh siswa yang ada pada SMA Negeri yang dipilih.



BAB 7

IMPLEMENTASI SISTEM

7.1. Model

1. Encryption Key

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class EncryptionKey extends Model
{
    protected $fillable = ['id', 'key',
'jadwal_ujian_id', 'created_at',
'updated_at'];

    public function jadwal_ujian()
    {
        return $this-
>belongsTo('App\Models\JadwalUjian');
    }
}
```

2. Jadwal Ujian

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class JadwalUjian extends Model
{
    /**
```

```

    * Config this model
    * jika incrementing false maka kita
    pakai UUID
    */
    protected $primaryKey='id';
    public $incrementing = false;

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'jadwal_ujian';

    /**
     * The attributes that are mass
    assignable.
     *
     * @var array
     */
    protected $fillable =[
        'id',
        'deskripsi',
        'sesi',
        'waktu_mulai',
        'waktu_selesai',
        'durasi',
        'tipe_ujian',
        'pelajaran_id'
    ];

    /**
     * One to Many relation
     *
     * @return Illuminate\Database\Eloquent\Relations\HasMany
     */
    public function ujianSiswas(){

```

```

        return $this->hasMany('App\Models\UjianSiswa');
    }

    public function uploads() {
        return $this->hasMany('App\Models\Uploads');
    }

    public function pakets() {
        return $this->belongsToMany('App\Models\Paket', 'ujian_paket')->withTimestamps();
    }

    public function jawabanEssai()
    {
        return $this->hasManyThrough('App\Models\JawabanEssai',
            'App\Models\UjianSiswa', 'jadwal_ujian_id',
            'ujian_siswa_id', 'id', 'id');
    }

    public function encryption_key()
    {
        return $this->hasOne('App\Models\EncryptionKey');
    }

    public function pelajaran()
    {
        return $this->belongsTo('App\Models\Pelajaran');
    }
}

```

3. Kabupaten Kota

```
<?php
```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class KabKota extends Model
{
    protected $table = 'kab_kota';
}

```

4. Paket

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Paket extends Model
{
    /**
     * Config this model
     * jika incrementing false maka kita
     pakai UUID
     */
    protected $primaryKey='id';
    public $incrementing = false;

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'paket';

    /**

```

```

        * The attributes that are mass
assignable.
        *
        * @var array
        */
        protected $fillable = [
            'id',
            'nama',
            'deskripsi',
            'pelajaran_id'
        ];

        /**
         * One to Many relation
         *
         * @return
Illuminate\Database\Eloquent\Relations\HasM
any
        */
        public function soals() {
            return $this->
hasMany('App\Models\Soal');
        }

        /**
         * One to Many relation
         *
         * @return
Illuminate\Database\Eloquent\Relations\HasM
any
        */
        public function ujianSiswas() {
            return $this->
hasMany('App\Models\UjianSiswa');
        }

        /**
         * One to Many relation
         *

```

```

        *                                     @return
Illuminate\Database\Eloquent\Relations\Belongs
ngsTo
        */
        public function pelajaran(){
            return $this->
>belongsTo('App\Models\Pelajaran');
        }

        public function jadwalujians(){
            return $this->
>belongsToMany('App\Models\JadwalUjian','uj
ian_paket')->withTimestamps();
        }
    }

```

5. Pelajaran

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Pelajaran extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'pelajaran';

    /**
     * The attributes that are mass
    assignable.
     *
     * @var array
    */
}

```

```

        */
        protected $fillable =[
            'id',
            'nama',
            'jenjang',
            'kurikulum'
        ];

        /**
         * One to Many relation
         *
         * @return Illuminate\Database\Eloquent\Relations\HasMany
         */
        public function pakets(){
            return $this->hasMany('App\Models\Paket');
        }

        public function user(){
            return $this->hasMany('App\Models\User');
        }

        public function jadwalwals(){
            return $this->hasMany('App\Models\JadwalUjian');
        }
    }

```

6. Sekolah

```

<?php

namespace App\Models;

```



```

use Illuminate\Database\Eloquent\Model;

class Sekolah extends Model
{
    /**
     * Config this model
     */
    protected $primaryKey = 'id';
    public $incrementing = false;

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'sekolah';

    /**
     * The attributes that are mass
    assignable.
     *
     * @var array
     */
    protected $fillable = [
        'id',
        'nama',
        'alamat',
        'npsn',
        'kurikulum',
        'kode'
    ];

    /**
     * One to Many relation
     *
     * @return
    Illuminate\Database\Eloquent\Relations\HasMany
     */
}

```

```

    public function users()
    {
        return $this->hasMany('App\Models\User');
    }

    public function server()
    {
        return $this->hasMany('App\Models\Server');
    }

    public function ujians()
    {
        return $this->hasManyThrough('App\Models\UjianSiswa',
            'App\Models\User');
    }

    public function namaSekolahCleaned()
    {
        return preg_replace('/[^\da-z]/i',
            '', $this->nama);
    }

    public function serial_numbers(){
        return $this->hasMany('App\Models\SerialNumber');
    }
}

```

7. Serial Number

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

```

```

class SerialNumber extends Model
{
    //
    public function sekolah()
    {
        return $this->
belongsTo('App\Models\Sekolah');
    }
}

```

8. Settings

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Settings extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'settings';

    /**
     * The attributes that are mass
    assignable.
     *
     * @var array
     */
    protected $fillable =[
        'id',
        'value'
    ];
}

```

```
}
```

9. Soal

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Soal extends Model
{
    /**
     * Config this model
     */
    protected $primaryKey='id';
    public $incrementing = false;

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'soal';

    /**
     * The attributes that are mass
    assignable.
     *
     * @var array
     */
    protected $fillable =[
        'id',
```

```

        'deskripsi',
        'pilihan_a',
        'pilihan_b',
        'pilihan_c',
        'pilihan_d',
        'pilihan_e',
        'kunci_jawaban',
        'tipe_soal',
        'bobot_essai',
        'tingkat_kesulitan',
        'level_kognitif',
        'paket_id',
        'anchor_number'
    ];

    /**
     * One to Many relation
     *
     * @return
     Illuminate\Database\Eloquent\Relations\HasMany
    */
    public function latexs() {
        return $this->hasMany('App\Models\Latex');
    }

    /**
     * One to Many relation
     *
     * @return
     Illuminate\Database\Eloquent\Relations\BelongsToMany
    */
    public function paket() {
        return $this->belongsToMany('App\Models\Paket');
    }

```

```

    public function jawabanEssai()
    {
        return $this->hasMany('App\Models\JawabanEssai');
    }

    public function users()
    {
        return $this->hasMany('App\Models\KorektorSoal');
    }

    public function correctionCounter() {
        return $this->belongsTo('App\Models\SoalCorrectionsCounter', 'id');
    }
}

```

10. Ujian Siswa

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class UjianSiswa extends Model
{
    /**
     * Config this model
     * jika incrementing false maka kita
    pakai UUID
     */
    protected $primaryKey='id';
    public $incrementing = false;

    /**

```

```

    * The database table used by the model.
    *
    * @var string
    */
protected $table = 'ujian_siswa';

/**
 * The attributes that are mass
assignable.
 *
 * @var array
 */
protected $fillable =[
'id',
'random_soal',
'random_jawaban',
'jawaban_siswa',
'waktu_mulai',
'waktu_selesai',
'jumlah_benar',
'jumlah_salah',
'jumlah_kosong',
'nilai_uraian',
'paket_id',
'user_id',
'server_id',
'jadwal_ujian_id'
];

/**
 * One to Many relation
 *
 * @return Illuminate\Database\Eloquent\Relations\BelongsTo
 */
public function server(){
    return $this->belongsTo('App\Models\Server');
}

```

```

    }

    /**
     * One to Many relation
     *
     * @return Illuminate\Database\Eloquent\Relations\BelongsToMany
     */
    public function user(){
        return $this->belongsToMany('App\Models\User');
    }

    /**
     * One to Many relation
     *
     * @return Illuminate\Database\Eloquent\Relations\BelongsToMany
     */
    public function user(){
        return $this->belongsToMany('App\Models\User');
    }

    /**
     * One to Many relation
     *
     * @return Illuminate\Database\Eloquent\Relations\BelongsToMany
     */
    public function jadwalUjian(){
        return $this->belongsToMany('App\Models\JadwalUjian');
    }

    /**

```



```

        * One to Many relation
        *
        *
        Illuminate\Database\Eloquent\Relations\BelongsToMany
        */
        public function jawabanEssai() {
            return $this->hasMany('App\Models\JawabanEssai');
        }

        public function paket() {
            return $this->belongsTo('App\Models\Paket');
        }

        public function soal($id)
        {
            $soal = Soal::find($id);
            return $soal;
        }
    }
}

```

11. User

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Uploads extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'uploads';
}

```

```

    /**
     * The attributes that are mass
    assignable.
     *
     * @var array
     */
    protected $fillable = [
        'id',
        'jenis',
        'nama_file',
        'path_file'
    ];

    public function sekolah() {
        return $this->
        belongsTo('App\Models\Sekolah');
    }

    public function jadwalUjian() {
        return $this->
        belongsTo('App\Models\JadwalUjian');
    }
}

```

7.2. Controller

1. Auth Controller

```

<?php

namespace App\Http\Controllers;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;
use App\Models\UjianSiswa;
use App\Models\User;

```

```

use App\Models\Sekolah;
use App\Models\EncryptionKey;
use Carbon\Carbon;
use Hash;
use File;

class AuthController extends Controller
{
    public function index()
    {
        return view('login');
    }

    public function login(Request $request){
        $users = User::where('username',
        $request->input('username'))->get();

        foreach ($users as $user)
        {
            if(Hash::check($request-
            >input('password'),$user->password)) {
                Auth::loginUsingId($user-
                >id, TRUE);
                if(Auth::user()->level ===
                'proktor')
                {
                    $log_file =
                    base_path('storage/logs/proktor_log.txt');
                    $content =
                    '['.Carbon::now('Asia/Jakarta')->format('d-
                    m-y H:i:s').'] '.$request->ip(). " - LOGGED
                    IN \n";
                    File::append($log_file,
                    $content);
                }
                return redirect('/home');
            }
        }
    }
}

```

```

        return redirect('/')-
>with('error','Username      atau      Password
salah!!');
    }

    public function home()
    {
        $data['user'] = Auth::user();
        $data['jadwals'] =
EncryptionKey::where('is_announce', 'Yes')-
>get();

        switch(Auth::user()->level){
            case 'admin':
                return
view('pages.admin.index', $data);
                break;
            case 'guru':
                return
view('pages.guru.index', $data);
                break;
            case 'proktor':
                $data['siswas'] =
User::where('sekolah_id',$data['user']-
>sekolah_id)->where('level','siswa')-
>get();
                $data['sekolah'] =
Sekolah::where('id',$data['user']-
>sekolah_id)->first();
                return
view('pages.proktor.index', $data);
                break;
            case 'korektor':
                return
view('pages.korektor.index', $data);
                break;
            default:
                Auth::logout();

```

```

        return redirect('/')-
>with('error','user tidak dikenali');
    }
}

public function logout(Request $request)
{
    $user = Auth::User();
    Auth::logout();
    if($user->level == 'siswa'){
        $ujiansiswas =
UjianSiswa::where('user_id', $user->id)-
>get();
        if(count($ujiansiswas))
        {
            $ujiansiswa = $ujiansiswas-
>first();
            $ujiansiswa->status = 0;
            $ujiansiswa->save();
        }
        return redirect('/')-
>with('success','logout berhasil');
    }
    if($user->level == 'proktor' ||
$user->level == 'admin' || $user->level ==
'guru' || $user->level == 'korektor'){
        $log_file =
base_path('storage/logs/proktor_log.txt');
        $content =
'['.Carbon::now('Asia/Jakarta')->format('d-
m-y H:i:s').'] '.$request->ip()." - LOGGED
OUT \n";
        File::append($log_file,
$content);
        return redirect('/')-
>with('success','logout berhasil');
    }
}

```

```

        public function indexGuruGantiPassword()
        {
            return
view('pages.guru.ganti_password');
        }
        public function
guruGantiPassword(Request $request)
        {
            $guru = User::find(Auth::user()-
>id);
            $guru->password = bcrypt($request-
>password);
            $guru->save();
            return redirect()->back()-
>with('success', 'Berhasil Ubah Password');
        }
    }
}

```

2. Enkripsi Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\EncryptionKey;
use App\Models\JadwalUjian;

class EnkripsiController extends Controller
{
    public function getKeyByJadwal($id)
    {
        $key
EncryptionKey::where('jadwal_ujian_id',
$id)->firstOrFail();
        return response()->json($key);
    }
}

```

```

public function update(Request $request)
{
    if($request->type == "edit") {
        $jadwal_ujian =
JadwalUjian::findOrFail($request-
>input('id'));
        $key = $jadwal_ujian-
>encryption_key;
        if($key)
            $key->key = $request-
>input('key');
        else
        {
            $key = new EncryptionKey;
            $key->key = $request-
>input('key');
            $key->jadwal_ujian_id =
$jadwal_ujian->id;
        }

        $key->save();

        return back()->with('success',
'Berhasil merubah password enkripsi');
    } else if ($request->type ==
"delete"){
        $jadwal_ujian =
JadwalUjian::findOrFail($request-
>input('id'));

        $new =
EncryptionKey::where('jadwal_ujian_id',
$request->input('id'))->first();

        if($new) {
            $new->is_announce = NULL;
            $new->save();
        }
    }
}

```

```

        return back()-
>with('success', 'Berhasil menyembunyikan
password enkripsi');
    } else {
        return back()->with('error',
'Gagal menyembunyikan password enkripsi');
    }

    } else {
        $new =
EncryptionKey::where('jadwal_ujian_id',
$request->input('id'))->first();
        if($new) {
            $new->is_announce = 'Yes';
            $new->save();
            return back()-
>with('success', 'Berhasil share password
enkripsi');
        } else {
            return back()->with('error',
'Gagal share password enkripsi');
        }
    }
}
}
}

```

3. Hasil Controller

```

<?php

namespace App\Http\Controllers;

use Carbon\Carbon;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;
use App\Models\JadwalUjian;

```



```

use App\Models\Soal;
use App\Models\KabKota;
use App\Models\Sekolah;
use App\Models\User;
use DB;
use App\Models\Uploads;
use Symfony\Component\Process\Process;
use
Symfony\Component\Process\Exception\Process
FailedException;

class HasilController extends Controller
{
    public function index()
    {
        $now = Carbon::now('Asia/Jakarta');
        $data['jadwals'] =
JadwalUjian::where('waktu_selesai', '<', $now
)->orderBy('waktu_selesai', 'desc')-
>paginate(5);
        return
view('pages.proktor.hasil.index', $data);
    }

    public function show($id)
    {
        $data['jadwal_ujian'] =
JadwalUjian::find($id);
        $data['ujian_siswas'] =
$data['jadwal_ujian']->ujianSiswas;
        return
view('pages.proktor.hasil.show', $data);
    }

    public function koreksi($id)
    {
        $jadwal_ujian =
JadwalUjian::find($id);

```

```

        $ujian_siswas      =      $jadwal_ujian-
>ujianSiswas;
        foreach($ujian_siswas      as
$ujian_siswa){
                $soals      =
json_decode($ujian_siswa->random_soal);
                $jawaban_posisi      =
json_decode($ujian_siswa->random_jawaban);
                $jawabans      =
json_decode($ujian_siswa->jawaban_siswa);
                $jawaban_benar = 0;
                $jawaban_kosong = 0;
                $jawaban_salah = 0;
                foreach($soals      as      $key      =>
$soal_id){
                        $soal      =
Soal::find($soal_id);
                        if($soal->tipe_soal      ==
"pilihan_ganda"){
                                if($jawabans[$key] == "
") {
                                        $jawaban_kosong+=1;
                                }else{
                                        $inttochar      =
array("1"      =>
"a", "2"=>"b", "3"=>"c", "4"=>"d", "5"=>"e");
                                        $arr_random      =
array("a"
$inttochar[$jawaban_posisi[$key][0]],
                                "b" => $inttochar[$jawaban_posisi[$key][1]],
                                "c" => $inttochar[$jawaban_posisi[$key][2]],
                                "d" => $inttochar[$jawaban_posisi[$key][3]],
                                "e"      =>
$inttochar[$jawaban_posisi[$key][4]]);

```

```

if($arr_random[strtolower($jawabans[$key])]
== strtolower($soal->kunci_jawaban)){

$jawaban_benar+=1;
    }else{

$jawaban_salah+=1;
    }
    }
    }

    $ujian_siswa->jumlah_benar =
$jawaban_benar;
    $ujian_siswa->jumlah_salah =
$jawaban_salah;
    $ujian_siswa->jumlah_kosong
= $jawaban_kosong;
    $ujian_siswa->save();
    }
    }
    $jadwal_ujian->status =
'Terkoreksi';
    $jadwal_ujian->save();
    return redirect()->back();
}

public function download(){
    return response()-
>download(storage_path('app/hasil.xlsx'));
}

public function sekolah($id)
{
    $kode_kota = '__'.$id.'%';
    $data['kota'] =
KabKota::where('kd_rayon', $id)->first();

```

```

        $data['sekolah'] =
Sekolah::where('kode', 'LIKE', $kode_kota) -
>get();

        $data['no'] = 1;
        return
view('pages.admin.hasil.sekolah', $data);
    }

    public function all($ujian, $id_sekolah)
    {
        if ($ujian == "all") {
            $data['hasil'] =
DB::select('SELECT *, us.nama, p.deskripsi
AS mapel FROM ujian_siswa uj JOIN user us ON
uj.user_id = us.id JOIN jadwal_ujian p ON
uj.jadwal_ujian_id = p.id WHERE uj.user_id
IN (SELECT id FROM user u WHERE
u.sekolah_id= :id) ORDER BY
uj.jadwal_ujian_id',[
                'id' => $id_sekolah
            ]);
        }
        else {
            $data['hasil'] =
DB::select('SELECT *, us.nama, p.deskripsi
AS mapel FROM ujian_siswa uj JOIN user us ON
uj.user_id = us.id JOIN jadwal_ujian p ON
uj.jadwal_ujian_id = p.id WHERE uj.user_id
IN (SELECT id FROM user u WHERE
u.sekolah_id= :id) and uj.jadwal_ujian_id
= :ujian ORDER BY uj.jadwal_ujian_id',[
                'id' => $id_sekolah,
                'ujian' => $ujian
            ]);
        }
        $data['ujian'] =
JadwalUjian::whereDate('waktu_selesai', '<',
Carbon::now()->format('Y-m-d'))-

```

```

>where('deskripsi', 'not like',
"%|OFFLINE|%" )->get();

        $data['sekolah'] =
Sekolah::where('id',$id_sekolah)->first();
        $data['no'] = 1;
        return view('pages.admin.hasil.all',
$data);
    }
}

```

4. Import Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Uploads;

class ImportController extends Controller
{
    public function index()
    {
        $data['pelajaran'] =
Uploads::where('jenis','pelajaran')->first();
        $data['sekolah'] =
Uploads::where('jenis','sekolah')->first();
        $data['siswas'] =
Uploads::where('jenis','siswa')->get();
        $data['nosiswa'] = 1;
        $data['ujians'] =
Uploads::where('jenis','ujian')->get();
        $data['noujian'] = 1;
        return
view('pages.proktor.import.index', $data);
    }
}

```

```

public function
importDataToDatabase(Request $request)
{
    $file_path = 'public/zipfile';
    $file_name = $request['file']-
>getClientOriginalName();
    $path = $request->file('file')-
>storeAs('public/zipfile', $file_name);

    $path_file = '/var/www/html/usek-
2017/storage/app/public/zipfile/' . $file_nam
e;

    $path_file2 =
'/tmp/mysql/' . $file_name;

    $command = 'sudo mv ' . $path_file . '
' . $path_file2;
    $command = escapeshellcmd($command);
    $output = shell_exec($command);

    $command = 'sudo chown mysql:mysql
' . $path_file2;
    $command = escapeshellcmd($command);
    $output = shell_exec($command);

    $output = "";
    $jenis = "";
    if (strpos($file_name,
'sekolah') !== false) {
        $jenis = 'sekolah';
        $command = 'sudo python
/var/www/html/usek-
2017/script_python/unpak_package.py master1
' . $file_name;
        $command =
escapeshellcmd($command);
        $output = shell_exec($command);

```

```

    } else if (stripes($file_name,
'pelajaran') != false) {
        $jenis = 'pelajaran';
        $command = 'sudo python
/var/www/html/usek-
2017/script_python/unpak_package.py master1
' . $file_name;
        $command =
escapeshellcmd($command);
        $output = shell_exec($command);
    } else if (stripes($file_name,
'siswa') != false) {
        $jenis = 'siswa';
        $command = 'sudo python
/var/www/html/usek-
2017/script_python/unpak_package.py master2
' . $file_name;
        $command =
escapeshellcmd($command);
        $output = shell_exec($command);
    } else {
        $jenis = 'ujian';
        $command = 'sudo python
/var/www/html/usek-
2017/script_python/unpak_package.py master3
' . $file_name;
        $command =
escapeshellcmd($command);
        $output = shell_exec($command);
    }

    $command = 'rm ' . $path_file2;
    $command = escapeshellcmd($command);
    $output = shell_exec($command);

    $status = TRUE; // TODO cek output
python script

    if($status)

```

```

        {
            $this->create($file_name,
$jenis);
            return redirect('/import')->with('Import Berhasil');
        }
        else
            return redirect('/import')->with('Import Gagal');
    }

    protected function create(String
$nama_file, String $jenis) {
        $upload = new Uploads();
        $upload->nama_file = $nama_file;
        $upload->jenis = $jenis;
        $upload->save();
    }

    public function reset() {
        $id = request('id');
        //TODO validation ujian

        Uploads::find($id)->delete();

        return response()->json($id);
    }
}

```

5. Jadwal Ujian Controller

```

<?php

namespace App\Http\Controllers;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;
use App\Models\JadwalUjian;
use App\Models\JadwalUjianServer;

```



```

use Uuid;
use PDF;
use DB;
use App\Models\Paket;
use App\Models\Server;
use App\Models\Pelajaran;
use App\Models\Soal;
use App\Models\Sekolah;
use App\Models\UjianSiswa;
use App\Models\User;
use App\Models\Uploads;
use App\Models\EncryptionKey;
use App\Models\MonitoringServer;
use App\Models\SerialNumber;
use Carbon\Carbon;

class JadwalUjianController extends Controller
{
    // menampilkan seluruh data jadwal
    public function index()
    {
        $pelajarans = Pelajaran::all();
        return
view('pages.proktor.ujian.jadwal', compact('
pelajarans'));
    }

    //memamsukkan data jadwal ke database
    public function storeJadwal(Request
$request)
    {
        $data = $request->except('_token',
'key');
        $data['id'] = Uuid::generate()-
>string;
        DB::beginTransaction();
        try {

```

```

        $jadwal =
JadwalUjian::create($data);
        $data_key = [
            'key' => $request-
>input('key'),
            'jadwal_ujian_id' =>
$jadwal->id
        ];

EncryptionKey::create($data_key);
        DB::commit();
    } catch (\Exception $e) {
        DB::rollback();
        dd($e);
        return back()->with('error',
'Gagal menambahkan jadwal ujian');
    }
    return
redirect('/jadwal/pilih_paket/' . $data['id']
);
}

    public function storeJadwalMany(Request
$request)
    {
        $data = $request->except('_token',
'key', 'number');
        try {
            for ($i=0; $i < $request-
>number; $i++) {
                DB::beginTransaction();
                $data['id'] =
Uuid::generate()->string;
                $jadwal =
JadwalUjian::create($data);
                DB::commit();
            }
        } catch (\Exception $e) {
            DB::rollback();

```

```

        dd($e);
        return back()->with('error',
        'Gagal menambahkan jadwal ujian');
    }
    return redirect()->back()-
>with('success','Berhasil Membuat Jadwal');
}

//merubah data jadwal yang ada didatabase
public function updateJadwal(Request
$request)
{
    $data = $request->all();
    $jadwal =
JadwalUjian::find($data['id']);
    $jadwal->deskripsi =
$data['deskripsi'];
    $jadwal->sesi = $data['sesi'];
    $jadwal->pelajaran_id =
$data['pelajaran_id'];
    $jadwal->waktu_mulai =
$data['waktu_mulai'];
    $jadwal->waktu_selesai =
$data['waktu_selesai'];
    $jadwal->durasi = $data['durasi'];
    $jadwal->save();

    return
    redirect('/jadwal/pilih_paket/'.$data['id']
    );
}
public function
updateWaktuJadwal(Request $request)
{
    $data = $request->all();
    $jadwal =
JadwalUjian::find($data['id']);
    $jadwal->waktu_mulai =
$data['waktu_mulai'];

```

```

        $jadwal->waktu_selesai =
$data['waktu_selesai'];
        $jadwal->durasi = $data['durasi'];
        $jadwal->save();

        $ujians = $jadwal->ujianSiswas;
        foreach ($ujians as $ujian)
        {
            $ujian->waktu_mulai = $jadwal-
>waktu_mulai;
            $ujian->waktu_selesai = $jadwal-
>waktu_selesai;
            $ujian->durasi = $jadwal-
>durasi;
            $ujian->save();
        }

        return redirect('/jadwal');
    }

    //untuk ajax mendapatkan jadwal
    public function getJadwal(Request
$request)
    {
        $id = $request->input('id');
        $jadwal = JadwalUjian::find($id);

        return response()->json($jadwal);
    }

    public function getSiswa(Request
$request)
    {
        $id = $request->input('id');
        $jadwal_ujian =
JadwalUjian::find($id);
        foreach($jadwal_ujian->ujianSiswas
as $key => $ujian_siswa){
            $nestedData = array();

```

```

        array_push($nestedData,$key+1);

array_push($nestedData,$ujian_siswa->user-
>username);
        $data[] = $nestedData;
    }
    $json_data =array(
        "data" => $data
    );
    return response()->json($json_data);
}

public function getJadwalServer(Request
$request)
{
    $id = $request->input('id');
    $jadwal = JadwalUjian::find($id);

    return response()->json($jadwal);
}

//untuk hapus data jadwal yang ada
didatabase (ajax)
public function deleteJadwal(Request
$request)
{
    $id = $request->input('id');
    $jadwal = JadwalUjian::find($id);
    $jadwal->pakets()->detach();
    $jadwal->encryption_key()-
>delete();
    $jadwal->delete();
    return response()->json($id);
}

public function deletePaketUjian(Request
$request)
{
    $id = $request->input('id');

```

```

        $ujians =
        UjianSiswa::where('jadwal_ujian_id', $id)-
        >get();

        foreach ($ujians as $ujian) {
            $ujian->delete();
        }
        return response()->json("sukses");
    }

    public function ajaxDatatables(Request
$request)
    {
        //Isi dengan kolom
        $columns = array(
            0 => 'deskripsi',
            1=> 'sesi',
            2=> 'waktu_mulai',
            3=> 'waktu_selesai',
            4=> 'durasi',
            5=> 'jumlah_upload',
            6=> 'key',
            7=> 'tipe_ujian',
            8=> 'action'
        );

        //<-- Gak Perlu Diubah -->
        $totalData = JadwalUjian::count();

        $totalFiltered = $totalData;

        $limit = $request->input('length');
        $start = $request->input('start');
        $order = $columns[$request-
>input('order.0.column')];
        $dir = $request-
>input('order.0.dir');

        if(empty($request-
>input('search.value')))

```

```

        {
            $jadwals =
JadwalUjian::offset($start)
                        ->limit($limit)
                    -
>orderBy('deskripsi','ASC')
                    -
>orderBy($order,$dir)
                        ->get();
        }
        else {
            $search = $request->
input('search.value');

            //<-- Gak Perlu Diubah END -->
            //Tambah orWhere di kolom yang akan
            dijadikan di search
            $jadwals =
JadwalUjian::where('deskripsi','LIKE',"%{$s
earch}%")
                        //
>orWhere('title', 'LIKE',"%{$search}%")
                    -
>offset($start)
                        ->limit($limit)
                    -
>orderBy('deskripsi','ASC')
                    -
>orderBy($order,$dir)
                        ->get();
            //Tambah orWhere di kolom yang akan
            dijadikan di search
            $totalFiltered =
JadwalUjian::where('deskripsi','LIKE',"%{$s
earch}%")
                        //
>orWhere('title', 'LIKE',"%{$search}%")
                        ->count();
        }

```

```

        $data = array();
        if(!empty($jadwals))
        {
            foreach ($jadwals as $jadwal)
            {
                //Apa yang akan ditampilkan
                di tiap2 row
                $nestedData['deskripsi'] =
                $jadwal->deskripsi.'<br>';
                foreach($jadwal->pakets as
                $paket){

                $nestedData['deskripsi'] =
                $nestedData['deskripsi'].'<span class="badge
                bg-green">'. $paket->nama.'</span>      <span
                class="badge                                bg-red"
                id="announced"></span>';
                }

                $nestedData['key'] = null;
                $key = $jadwal->
                encryption_key;
                if($key) {
                    $nestedData['key'] =
                    $key->key;
                    if($key->is_announce ==
                    'Yes') {

                    $nestedData['deskripsi'] =
                    $nestedData['deskripsi'].'<span class="badge
                    bg-red">Announced</span>';
                    }
                }
                $nestedData['sesi'] =
                $jadwal->sesi;

                // ID setter, untuk
                keperluan ajax delete

```



```

$nestedData['DT_RowId'] =
$jadwal->id;

$nestedData['waktu_mulai'] =
$jadwal->waktu_mulai;

$nestedData['waktu_selesai'] = $jadwal-
>waktu_selesai;

$nestedData['durasi'] =
$jadwal->durasi;

$nestedData['jumlah_upload'] = $jadwal-
>uploads->groupBy('sekolah_id')->count();

// if($jadwal->tipe_ujian ==
1) $tipex = 'Offline';
// else if($jadwal-
>tipe_ujian == 2) $tipex = 'Online';
// $tipex = $jadwal-
>tipe_ujian;

// $nestedData['tipe_ujian']
= $tipex;

$nestedData['action'] =
'<button class="btn bg-blue
edit" data-toggle="modal" data-
target="#modalEdit"
onClick="editFunction(\''. $jadwal-
>id.\')"><i class="material-
icons">mode_edit</i></button>

<button class="btn bg-red
delete" data-toggle="modal" data-
target="#modalKonfirmasiDelete"
onClick="deleteClickFunction(\''. $jadwal-
>id.\')" ><i class="material-
icons">delete</i></button>

```

```

                                <button class="btn bg-yellow
edit"                        data-toggle="modal"      data-
target="#modalInfo"
onClick="moreInfoFunction(\''.$jadwal-
>id.'\')"><i                        class="material-
icons">info</i></button> ';
                                if(auth()->user()->level ===
'admin'){
                                $nestedData['action'] =
$nestedData['action'].' <a class="btn bg-
blue-grey"
href="/jadwal/monitoring/'.$jadwal-
>id.'"><i                        class="material-
icons">screen_share</i></a>
                                <a class="btn bg-purple"
href="/hasil/'.$jadwal->id.'"><i
class="material-
icons">folder_shared</i></a>
                                <button class="btn bg-
green"                        data-toggle="modal"      data-
target="#modalEditKey"
onClick="editKeyFunction(\''.$jadwal-
>id.'\')"><i                        class="material-
icons">enhanced_encryption</i></button>';
                                }
                                $data[] = $nestedData;
                                }
                                }

//<-- Gak Perlu Diubah -->
$json_data = array(
                                "draw"                                =>
intval($request->input('draw')),
                                "recordsTotal"                        =>
intval($totalData),
                                "recordsFiltered"                    =>
intval($totalFiltered),

```

```

                                "data"                                ==>
$data
                                );

                                return json_encode($json_data);
                                //<-- Gak Perlu Diubah END -->
                                }

                                public function showPackets($jadwal_id)
                                {
                                    $jadwal                                =
JadwalUjian::find($jadwal_id);
                                    $data['jadwalUjian'] = $jadwal;
                                    $data['pelajarans']   =
Pelajaran::all();
                                    return
view('pages.proktor.ujian.pilih_paket',
$data);
                                }

                                public function choosePackets(Request
$request, $jadwal_id)
                                {
                                    $pakets = $request->input('pakets');
                                    $jadwal                                =
JadwalUjian::findOrFail($jadwal_id);
                                    $jadwal->pakets()->detach();
                                    if($pakets){
                                        foreach ($pakets as $paket)
                                        {
                                            $jadwal->pakets()-
>attach($paket);
                                        }
                                    }

                                }

                                return                                redirect('/jadwal')-
>with('success', 'Berhasil menambah jadwal
ujian');
                                }

```

```

        public function showSiswas($jadwal_id){
            $data['jadwalUjian'] =
JadwalUjian::find($jadwal_id);
            $data['sekolahs'] = Sekolah::all();
            return
view('pages.proktor.ujian.pilih_siswa',$data);
        }

        public function chooseSiswas(Request
$request, $jadwal_id)
        {
            $siswas = $request->input('siswas');
            $jadwal =
JadwalUjian::findOrFail($jadwal_id);

            if($siswas)
            {
                foreach($siswas as $siswa)
                {
                    $data['user_id'] = $siswa;
                    $data['id'] =
Uuid::generate()->string;
                    $data['paket_id'] = "0";
                    $data['jadwal_ujian_id'] =
$jadwal_id;
                    $data['waktu_mulai'] =
$jadwal->waktu_mulai;
                    $data['waktu_selesai'] =
$jadwal->waktu_selesai;
                    $data['durasi'] = $jadwal-
>durasi;
                    $ujian_siswa =
UjianSiswa::create($data, ['except' =>
['_token', 'submit']]);
                }
            }
        }

```

```

        return
    redirect ('/jadwal/random/' . $jadwal_id);
    }

    public function random($jadwal_id) {
        $jadwal_ujian =
    JadwalUjian::find($jadwal_id);
        $pakets = $jadwal_ujian->pakets;
        $ujian_siswas = $jadwal_ujian-
    >ujianSiswas;
        foreach($ujian_siswas as
    $ujian_siswa) {
            $i = rand(0, $pakets->count()-1);
            $paket = $pakets[$i];
            $arr_soal_id_with_essai =
    array();

            //pilgan
            $soals = $paket->soals-
    >where('tipe_soal', 'pilihan_ganda');
            $arr_soal_id = array();
            $arr_random_jawaban = array();
            $arr_jawaban =
    array("1", "2", "3", "4", "5");
            $arr_jawaban_siswa = array();
            foreach($soals as $soal) {
                shuffle($arr_jawaban);

    array_push($arr_soal_id, $soal->id);

    array_push($arr_random_jawaban, $arr_jawaban
    );

    array_push($arr_jawaban_siswa, " ");
            }
            shuffle($arr_soal_id);
            // dd($arr_soal_id);
            foreach ($arr_soal_id as
    $soal_id) {

```

```

array_push($arr_soal_id_with_essai,
$soal_id);
    }
    // dd($arr_soal_id_with_essai);

    //essai
    $soals_essai = $paket->soals-
>where('tipe_soal','essai');
    foreach ($soals_essai as
$soal_essai) {

array_push($arr_soal_id_with_essai,
$soal_essai->id);
    }
    // dd($arr_soal_id_with_essai);

    $ujian_siswa->paket_id = $paket-
>id;
    $ujian_siswa->random_soal =
json_encode($arr_soal_id_with_essai);
    $ujian_siswa->random_jawaban =
json_encode($arr_random_jawaban);
    $ujian_siswa->jawaban_siswa =
json_encode($arr_jawaban_siswa);
    $ujian_siswa->save();
    }
    // dd($ujian_siswas[0]);
    return redirect('/jadwal');
}

public function monitoring($id)
{
    $jadwal = JadwalUjian::find($id);
    return
view('pages.admin.upload.monitoring',
compact('jadwal', 'uploads'));
}

```

```

        public function
datatablesUploadMonitoring(Request $request)
    {
        // format : 5->idujian 6->idsekolah
        //Isi dengan kolom
        $columns = array(
            0 => 'sekolah_id',
            1 => 'nama_file',
            2 => 'server',
            3 =>
'tanggal_ujian',
            4 => 'created_at'
        );

        //<-- Gak Perlu Diubah -->
        $totalData =
Uploads::where('jadwal_ujian_id', $request-
>input('id'))->count();
        $totalFiltered = $totalData;

        $limit = $request->input('length');
        $start = $request->input('start');
        $order = $columns[$request-
>input('order.0.column')];
        $dir = $request-
>input('order.0.dir');

        if(empty($request-
>input('search.value')))
        {
            $uploads =
Uploads::where('jadwal_ujian_id', $request-
>input('id'))
                ->offset($start)
                ->limit($limit)
                -
>orderBy($order,$dir)
                ->get();

```

```

    }
    else {
        $search = $request-
>input('search.value');
        //<-- Gak Perlu Diubah END -->
        //Tambah orWhere di kolom yang akan
        dijadikan di search
        $uploads =
Uploads::where('jadwal_ujian_id', $request-
>input('id'))

-
>where(function ($q) use ($search){
        $q-
>where('nama_file','LIKE','%{$search}%");
        $q-
>orWhereHas('sekolah', function($query) use
($search){
        $query-
>where('nama', 'LIKE', "%{$search}%");
        });
        $q-
>orWhereHas('jadwalUjian', function($query)
use ($search){
        $query-
>where('waktu_mulai', 'LIKE',
"%{$search}%");
        });
        $q-
>orWhere('created_at', 'LIKE',
"%{$search}%");
        })
-
>offset($start)
-
->limit($limit)
-
>orderBy($order,$dir)
        ->get();
        //Tambah orWhere di kolom yang akan
        dijadikan di search

```



```

                                $totalFiltered    =    $uploads->
>count();
    }

    $data = array();
    $data_uploads = $uploads;
    if(!empty($data_uploads))
    {
        foreach ($data_uploads as $index
=> $upload)
        {
            $nestedData['sekolah_id'] =
$upload->sekolah->nama;
            $ujian    =    $upload->
>jadwalUjian;
            $nestedData['nama_file'] =
$upload->nama_file;
            $nestedData['server'] =
explode("_", $upload->nama_file)[6][0];

            $nestedData['tanggal_ujian'] = date('d/m/Y',
strtotime($ujian->waktu_mulai));
            $nestedData['created_at'] =
date('d/m/Y      H:i',      strtotime($upload->
>created_at->tz('Asia/Jakarta')));

            $data[] = $nestedData;
        }
    }

    //<-- Gak Perlu Diubah -->
    $json_data = array(
        "draw" =>
intval($request->input('draw')),
        "recordsTotal" =>
intval($totalData),
        "recordsFiltered" =>
intval($totalFiltered),

```

```

                                "data"                                =>
$data
                                );
                                return json_encode($json_data);
                                //<-- Gak Perlu Diubah END -->
                                }

                                public function checkPassword(Request
$request){
                                    $server_id = $request->server_id;

                                    $check                                =
SerialNumber::where('server_id',$server_id)
->exists();
                                    // return response()->json($check);
                                    if ($check) {
                                        $now                                =
Carbon::now('Asia/Jakarta');
                                        $data                                =
JadwalUjian::where('waktu_mulai','<',$now)-
>where('waktu_selesai','>',$now)-
>with('encryption_key')->get();
                                        if (count($data) > 0) {
                                            return                        response()-
>json($data);
                                        }
                                        else {
                                            return                        response()-
>json('no_current',404);
                                        }
                                    }
                                    else {
                                        return                        response()-
>json('unregistered_server',400);
                                    }
                                }

```

```

    public function izinUpdate(Request
$request){
        if (!$request->server_id) {
            return response() -
>json('not_permitted',401);
        }

        $server_id = $request->server_id;
        $serial_number
        SerialNumber::where('server_id',$server_id)
->first();
        if (!$serial_number) {
            return response() -
>json('not_found',400);
        }

        if ($serial_number->permission == 0)
        {
            return response() -
>json('not_permitted',401);
        }

        $query = array([
            'server_id' => $server_id,
            'data_awal' => $request-
>data_awal,
            'data_baru' => $request-
>data_baru,
            'created_at' => Carbon::now()
        ]);

        $db =
DB::table('log_perubahan_jadwal_ujian')-
>insert($query);
        if (!$db) {
            return response()->json('400');
        }

        // $serial_number->permission = 0;

```

```

        // $serial_number->save();

        return response()->json('200',200);
    }

    public function updateLog()
    {
        $log = DB::table('log_perubahan_jadwal_ujian')->get();
        return view('pages.admin.server.updatelog', compact('log'));
    }

    public function ajaxLog(Request $request){
        //Isi dengan kolom

        $columns = array(
            0 => 'server_id',
            1 => 'nama_ujian',
            2 => 'data_awal',
            3 => 'data_baru',
            4 => 'waktu_request',
        );

        //<-- Gak Perlu Diubah -->
        $totalData = DB::table('log_perubahan_jadwal_ujian')->count();

        $totalFiltered = $totalData;

        $limit = $request->input('length');
        $start = $request->input('start');
        $order = $columns[$request->input('order.0.column')];
    }

```

```

        $dir = $request->
>input('order.0.dir');

        if(empty($request->
>input('search.value')))
        {
            $logs =
DB::table('log_perubahan_jadwal_ujian')->
>offset($start)
            ->limit($limit)
            ->orderBy($order,$dir)
            ->get();
            // return response()-
>json('here',200);
        }
        else {
            $search = $request->
>input('search.value');
            //<-- Gak Perlu Diubah END -->
            //Tambah orWhere di kolom yang akan
            dijadikan di search
            $logs =
DB::table('log_perubahan_jadwal_ujian')->
>where('server_id','LIKE','"%{$search}%")
            // -
>orWhere('title','LIKE','"%{$search}%")
            -
>offset($start)
            ->limit($limit)
            -
>orderBy($order,$dir)
            ->get();
            //Tambah orWhere di kolom yang akan
            dijadikan di search
            $totalFiltered =
DB::table('log_perubahan_jadwal_ujian')->
>where('server_id','LIKE','"%{$search}%")
            // -
>orWhere('title','LIKE','"%{$search}%")

```

```

        ->count();
    }

    $data = array();
    if(!empty($logs))
    {
        foreach ($logs as $log)
        {
            //Apa yang akan ditampilkan
            di tiap2 row
            $nestedData['server_id'] =
            $log->server_id;
            $data_log =
            explode('|',$log->data_awal);
            $nestedData['nama_ujian'] =
            $data_log[0];
            $nestedData['data_awal'] =
            $data_log[1]. ' <strong>-</strong>
            '.$data_log[2];
            $data_log =
            explode('|',$log->data_baru);
            $nestedData['data_baru'] =
            $data_log[1]. ' <strong>-</strong>
            '.$data_log[2];

            $nestedData['waktu_request'] = $log->
            >created_at;

            $data[] = $nestedData;
        }
    }

    //<-- Gak Perlu Diubah -->
    $json_data = array(
        "draw" =>
        intval($request->input('draw')),
        "recordsTotal" =>
        intval($totalData),

```

```

                                "recordsFiltered" =>
intval($totalFiltered),
                                "data" =>
$data
                                );

        return json_encode($json_data);
        //<-- Gak Perlu Diubah END -->

    }
}

```

6. Kabupaten/Kota Controller

```

<?php

namespace App\Http\Controllers;

use App\Models\KabKota;

use Illuminate\Http\Request;

class KabKotaController extends Controller
{
    public function index(){
        $data['kota'] =
KabKota::orderBy('kd_rayon')->get();
        $data['no'] = 1;
        return
view('pages.admin.kabkota.index', $data);
    }

    public function limitDownload(Request
$request, $action){
        if ($action == 'akses') {
            $kabkota =
KabKota::find($request->kota_id);

```

```

        $kabkota->download = 1;
        $kabkota->save();
        return redirect()->back()-
>with('success','Izin berhasil diberikan');
    } else if ($action == 'cabut') {
        $kabkota
KabKota::find($request->kota_id);
        $kabkota->download = 0;
        $kabkota->save();
        return redirect()->back()-
>with('success','Izin berhasil diberikan');
    } else {
        return
        redirect(route('kabkota.index'));
    }
}
}

```

7. Monitoring Document Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Dokumen;
use App\Models\Pelajaran;
use Carbon\Carbon;
use File;
use Illuminate\Support\Facades\Response;

class MonitoringDocumentController extends
Controller
{
    public function index()
    {
        return
        view('pages.admin.dokumen.index');
    }
}

```



```

    }

    public function datatables(Request
$request)
    {
        //Isi dengan kolom
        $columns = array(
            0=>'created_at',
            1=>'jenis',
            2=>'pelajaran',
            3=>'sesi',
            4=>'server',
            5=>'ruangan',
            6=>'action'
        );

        //<-- Gak Perlu Diubah -->
        $totalData = Dokumen::count();

        $totalFiltered = $totalData;

        $limit = $request->input('length');
        $start = $request->input('start');
        $order = $columns[$request-
>input('order.0.column')];
        $dir = $request-
>input('order.0.dir');

        if(empty($request-
>input('search.value')))
        {
            $dokuments =
Dokumen::offset($start)
                ->limit($limit)
                -
>orderBy($order,$dir)
                ->get();
        }
        else {

```

```

                                $search          =          $request-
>input('search.value');

                                //<-- Gak Perlu Diubah END -->
                                //Tambah orWhere di kolom yang akan
                                dijadikan di search
                                $dokuments          =
Dokumen::where('jenis', 'LIKE', "%{$search}%"
)
                                -
>orWhere('sesi', 'LIKE', "%{$search}%")
                                -
>orWhere('server', 'LIKE', "%{$search}%")
                                -
>orWhere('ruangan', 'LIKE', "%{$search}%")
                                -
>orWhereHas('pelajaran', function($query) use
($search) {
                                $query-
>where('nama', 'LIKE', "%{$search}%");
                                })
                                -
>offset($start)
                                ->limit($limit)
                                -
>orderBy($order,$dir)
                                ->get();
                                //Tambah orWhere di kolom yang akan
                                dijadikan di search
                                $totalFiltered          =
Dokumen::where('jenis', 'LIKE', "%{$search}%"
)
                                -
>orWhere('sesi', 'LIKE', "%{$search}%")
                                -
>orWhere('server', 'LIKE', "%{$search}%")
                                -
>orWhere('ruangan', 'LIKE', "%{$search}%")

```

```

-
>orWhereHas('pelajaran',function($query) use
($search) {
                                $query-
>where('nama', 'LIKE', "%{$search}%");
                                })
                                ->count();
    }

    $data = array();
    if(!empty($dokumens))
    {
        foreach ($dokumens as $dokumen)
        {
            //Apa yang akan ditampilkan
di tiap2 row
            $nestedData['jenis'] =
$dokumen->jenis;
            $nestedData['sesi'] =
$dokumen->sesi;
            $nestedData['server'] =
$dokumen->server;
            $nestedData['ruangan'] =
$dokumen->ruangan;
            $nestedData['pelajaran'] =
$dokumen->pelajaran->nama;
            $nestedData['created_at'] =
$dokumen->created_at->format('d F Y H:i');
            $nestedData['action'] = '<a
href="/dokumen/monitor/download/'. $dokumen-
>id.'" class="btn btn-sm btn-primary">

<span>Download</span>

                                </a><button
class="btn                                bg-red"
onclick="showDokumen(\''. $dokumen-
>file.\', \\''. $dokumen->id.\')">

<span>Preview</span>

```

```

</button>' ; ;

        $data[] = $nestedData;
    }
}

    //<-- Gak Perlu Diubah -->
    $json_data = array(
        "draw" =>
intval($request->input('draw')),
        "recordsTotal" =>
intval($totalData),
        "recordsFiltered" =>
intval($totalFiltered),
        "data" =>
$data
    );

    return json_encode($json_data);
    //<-- Gak Perlu Diubah END -->
}

public function download($id)
{
    $dokumen = Dokumen::findOrFail($id);
    $path = $dokumen->getFileFullPath();
    return response()->download($path);
}

public function preview($id){
    $dokumen = Dokumen::findOrFail($id);

    $path = $dokumen->getFileFullPath();
    $file = File::get($path);
    $type = File::mimeType($path);

    $response = Response::make($file,
200);

```

```

        $response->header("Content-Type",
$type);

        return $response;
    }
}

```

8. Monitoring Server Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\MonitoringServer;
use App\Models\JadwalUjian;
use App\Models\LogServer;
use App\Models\Sekolah;
use Carbon\Carbon;
use DB;

class MonitoringServerController extends Controller
{
    public function setState(Request $request)
    {
        $data = $request->only(['server_id',
'jadwal_ujian_id', 'sekolah_id', 'state']);
        $jadwal =
JadwalUjian::find($data['jadwal_ujian_id'])
;
        $sekolah =
Sekolah::find($data['sekolah_id']);

        $data['deskripsi_jadwal'] =
$jadwal ? $jadwal->deskripsi : '-';
    }
}

```

```

        $data['sekolah_nama'] = $sekolah-
>nama;

        $server
MonitoringServer::where('jadwal_ujian_id',
$data['jadwal_ujian_id'])-
>where('server_id', $data['server_id'])-
>first();
        if($server) {
            //apabila server bawah ngesubmit
            finished tapi ternyata sudah upload
            if($data['state'] == 'Finished'
            && $server->state == 'Uploaded')
            {
                LogServer::create($data);
            //create log finished
                return response()-
>json('200');
            }
            else
            {
                //create new for history,
                update for monitoring
                if($server->update($data) &&
                LogServer::create($data)) {
                    return response()-
>json('200');
                } else {
                    return response()-
>json('500');
                }
            }
        } else {
            //create new

            if(MonitoringServer::create($data) &&
            LogServer::create($data)) {
                return response()-
>json('200');

```

```

        } else {
            return response() -
>json('500');
        }
    }
}

public function monitorServer()
{
    //belum filter ujian yang jalan hari
    ini
        $data['tanggal_ujians'] =
MonitoringServer::select(DB::raw('DATE(crea
ted_at) as date'))->groupBy('date')->get();
        foreach ($data['tanggal_ujians'] as
$key => $tanggal)
        {
            $date_now =
Carbon::parse($tanggal->date);
            $date_besoknya =
Carbon::parse($tanggal->date)->addDay();
            $data['tanggal_ujians'][$key]-
>jadwals = MonitoringServer::distinct()-
>where('created_at','>',$date_now)-
>where('created_at','<',$date_besoknya)-
>get(['jadwal_ujian_id',
'deskripsi_jadwal']);
        }
        return
view('pages.admin.server.monitoring',
$data);
    }

    public function getServerSekolah(Request
$request)
    {
        $sekolahs =
MonitoringServer::where('jadwal_ujian_id',

```

```

$request->jadwal_ujian_id)->distinct()-
>get(['sekolah_id', 'sekolah_nama']);
    return $sekolahs;
}

    public function getServer(Request
$request)
    {
        $servers =
MonitoringServer::where('jadwal_ujian_id',
$request->jadwal_ujian_id)-
>where('sekolah_id', $request->sekolah_id)-
>get();
        return $servers;
    }

    public function historyServer()
    {
        return
view('pages.admin.server.history');
    }

    public function
historyServerDatatables(Request $request)
    {
        //Isi dengan kolom
        $columns = array(
            0 => 'sekolah_nama',
            1 => 'server_id',
            2 =>
'deskripsi_jadwal',
            3 => 'state',
            4 => 'created_at'
        );

        //<-- Gak Perlu Diubah -->
        $totalData = LogServer::count();

        $totalFiltered = $totalData;

```



```

        $limit = $request->input('length');
        $start = $request->input('start');
        $order  = $columns[$request-
>input('order.0.column')];
        $dir    = $request-
>input('order.0.dir');

        if(empty($request-
>input('search.value')))
        {
            $logs =
LogServer::offset($start)
                                ->limit($limit)
                                -
>orderBy($order,$dir)
                                ->get();
        }
        else {
            $search = $request-
>input('search.value');

            //<-- Gak Perlu Diubah END -->
            //Tambah orWhere di kolom yang akan
            dijadikan di search
            $logs =
LogServer::where('sekolah_nama','LIKE',"%{$
search}%")
                                -
>orWhere('server_id', 'LIKE',"%{$search}%")
                                -
>orWhere('deskripsi_jadwal',
'LIKE',"%{$search}%")
                                -
>orWhere('state', 'LIKE',"%{$search}%")
                                -
>offset($start)
                                ->limit($limit)

```

```

>orderBy($order,$dir)
                                ->get();
                                //Tambah orWhere di kolom yang akan
                                dijadikan di search
                                $totalFiltered
LogServer::where('sekolah_nama','LIKE','"%{$search}%"')
                                -
>orWhere('server_id','LIKE','"%{$search}%"')
                                -
>orWhere('deskripsi_jadwal',
'LIKE','"%{$search}%"')
                                -
>orWhere('state','LIKE','"%{$search}%"')
                                ->count();
                                }

    $data = array();
    if(!empty($logs))
    {
        foreach ($logs as $jadwal)
        {
            //Apa yang akan ditampilkan
            di tiap2 row
            $nestedData['sekolah_nama']
= $jadwal->sekolah_nama;
            $nestedData['server_id'] =
$jadwal->server_id;

            $nestedData['deskripsi_jadwal'] = $jadwal->
deskripsi_jadwal;
            $nestedData['state'] =
$jadwal->state;
            $nestedData['created_at'] =
$jadwal->created_at->format('d F Y H:i');

            $data[] = $nestedData;
        }
    }

```

```

    }

    //<-- Gak Perlu Diubah -->
    $json_data = array(
        "draw" =>
intval($request->input('draw')),
        "recordsTotal" =>
intval($totalData),
        "recordsFiltered" =>
intval($totalFiltered),
        "data" =>
$data
    );

    return json_encode($json_data);
    //<-- Gak Perlu Diubah END -->
}

public function checkUpload(Request
$request)
{
    $server =
MonitoringServer::where('jadwal_ujian_id',
$request->jadwal_ujian_id)-
>where('server_id', $request->server_id)-
>first();
    if($server)
        return response()->json($server-
>state);
    else
        return response()->json('Not
Found');
}
}

```

9. Monitoring Ujian Controller

```
<?php
```

```

namespace App\Http\Controllers;

use Carbon\Carbon;
use Illuminate\Http\Request;
use App\Models\User;
use App\Models\UjianSiswa;
use App\Models\JadwalUjian;

class MonitoringUjianController extends Controller
{
    public function index()
    {
        $now = Carbon::now('Asia/Jakarta');
        $data['jadwals'] =
        JadwalUjian::where('waktu_mulai', '<', $now) -
        >where('waktu_selesai', '>', $now) -
        >paginate(3);
        // dd($data['jadwals'], $now);
        return
        view('pages.proktor.monitoring.index', $data
        );
    }

    public function ujianSiswa($id)
    {
        $siswas =
        UjianSiswa::where('jadwal_ujian_id', $id) -
        >get();
        $chunks = $siswas->chunk(6);
        $rows = count($chunks);
        // parse waktu ujian

        $jadwal_ujian =
        JadwalUjian::find($id);
        $waktu_mulai =
        Carbon::parse($jadwal_ujian-

```

```

>waktu_mulai,'Asia/Jakarta')-
>timestamp*1000;
    $waktu_selesai =
Carbon::parse($jadwal_ujian-
>waktu_selesai,'Asia/Jakarta')-
>timestamp*1000;
    $waktu_sekarang =
Carbon::now('Asia/Jakarta')-
>timestamp*1000;

    $jadwal_id = $id;
    return
view('pages.proktor.monitoring.ujiansiswa',
compact('chunks', 'rows', 'waktu_mulai',
'waktu_selesai', 'waktu_sekarang',
'jadwal_id'));
    }

    public function getUjianSiswa(Request
$request)
    {
        $data['ujian'] =
UjianSiswa::where('user_id', $request->id)-
>where('jadwal_ujian_id', $request-
>jadwal_id)->first();
        //$ujian =
UjianSiswa::where('user_id', $request->id)-
>first();
        $data['siswa'] =
User::find($request->id);
        return response()->json($data);
    }

    public function getStatusUjianSiswa(Request $request)
    {
        $siswas =
UjianSiswa::where('jadwal_ujian_id',
$request->id)->get();

```

```

        return response()->json($siswas);
    }
}

```

10. Paket Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Uuid;
use App\Models\Soal;
use App\Models\Paket;
use App\Models\Pelajaran;

class PaketController extends Controller
{
    /* Constructor
    - Only auth
    */
    public function __construct()
    {
        // $this->middleware('auth');
    }

    /* Read
    - Read
    tampilkan paket
    */
    public function index()
    {
        $user = Auth::user();
        $pakets =
        Paket::where('pelajaran_id',$user->pelajaran_id)->orderBy('created_at',
        'asc')->get();
    }
}

```

```

        foreach ($pakets as $paket) {
            $paket['jumlah_soal'] =
Soal::where('paket_id',$paket->id)-
>count();
        }
        $no = 1;
        $pelajarans = Pelajaran::all();
        return view('pages/guru/paket',
compact('pakets','no','pelajarans'));
    }

    /* Edit
    - Edit
        tampilan editing nya pake modal
        berupa ajax request
    */
    public function edit()
    {
        $data = request()->all();

        $paket =
Paket::find($data['paket_id']);

        $paket->nama = $data['nama'];
        $paket->deskripsi =
$data['deskripsi'];
        // $paket->pelajaran_id =
$data['pelajaran_id'];

        $paket->save();
        if(Auth::user()->level=='admin'){
            return
redirect('/paketsoal/'.$paket->
pelajaran_id);
        }
        return redirect('/paket');
    }

    public function create()

```

```

    {
        $data = request()->all();
        $data['id'] = Uuid::generate()-
>string;
        $paket = Paket::create($data,
['except' => ['_token', 'submit']]);

        return redirect()->action(
            'SoalController@index',['id' =>
$data['id']]
        );
    }

    public function delete()
    {
        $id = request('id');
        $paket = Paket::find($id);
        $paket->delete();
        $soals =
Soal::where('paket_id',$id);
        $soals->delete();

        return response()->json($id);
    }

    public function copy()
    {
        $id = request('id');
        $paket = Paket::find($id);

        $new_paket = $paket->replicate();
        $new_paket['id'] = Uuid::generate()-
>string;
        $new_paket->push();

        $soals =
Soal::where('paket_id',$id)->get();
        foreach ($soals as $soal) {
            $new_soal = $soal->replicate();

```



```

        $new_soal['id'] =
        Uuid::generate()->string;
        $new_soal['paket_id'] =
        $new_paket['id'];

        $new_soal->push();
    }

    public function getPaket()
    {
        $id = request('id');
        $paket = Paket::find($id);

        return response()->json($paket);
    }

    public function printPaket()
    {
        return redirect('/paket');
    }
}

```

11. Paket Soal Controller

```

<?php

namespace App\Http\Controllers;
use App\Models\Pelajaran;
use App\Models\Paket;
use App\Models\Settings;
use Illuminate\Support\Carbon;
use Illuminate\Http\Request;

class PaketSoalController extends Controller
{
    public function index()
    {

```

```

        $data['pelajarans'] =
Pelajaran::all();
        $data['limit'] =
Settings::where('id','batas_input_soal')-
>first();
        return
view('pages.admin.paket.index',$data);
    }

    public function detail($pelajaran_id)
    {
        $pakets =
Paket::where('pelajaran_id',$pelajaran_id)-
>orderBy('created_at','asc')->get();
        foreach ($pakets as $paket) {
            $paket['jumlah_soal'] = $paket-
>soals->count();
        }
        $no = 1;
        $pelajarans = Pelajaran::all();
        return
view('pages/admin/paket/detail',
compact('pakets','no','pelajarans'));
    }

    public function updateLimit(Request
$request)
    {
        $settings =
Settings::where('id','batas_input_soal')-
>first();
        $date = Carbon::createFromFormat('Y-
m-d', $request->batas_input_soal);
        $date->hour = 0;
        $date->minute = 0;
        $date->second = 0;

        $settings->update([
            'value' => $date

```

```

    });
    return redirect('/paketsoal')-
>with('success','Batas input soal telah
    dirubah');
    }
}

```

12. Pelajaran Controller

```

<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use App\Models\Pelajaran;
use App\Models\Paket;
use App\Models\Soal;
use App\Models\UjianSiswa;

class PelajaranController extends Controller
{
    public function index()
    {
        $data['pelajarans'] =
Pelajaran::all();
        return
view('pages.admin.pelajaran.index',$data);
    }

    public function create(Request $request)
    {
        $data = $request->all();
        $pelajaran =
Pelajaran::create($data,
['except'=>['_token']]);
        return redirect()->to('pelajaran')

```

```

        ->with('success',"Berhasil
Menambahkan Pelajaran");
    }

    public function get(Request $request)
    {
        $pelajaran =
Pelajaran::find($request->id);
        return response()->json($pelajaran);
    }

    public function update(Request $request)
    {
        $pelajaran =
Pelajaran::find($request->id);
        $pelajaran->nama = $request->nama;
        $pelajaran->jenjang = $request-
>jenjang;
        $pelajaran->jenjang = $request-
>jenjang;
        $pelajaran->kurikulum = $request-
>kurikulum;
        $pelajaran->save();
        return redirect()->to('pelajaran')
        ->with('success',"Berhasil
Update Pelajaran");
    }

    public function delete(Request $request)
    {
        $pelajaran =
Pelajaran::find($request->id);
        $pelajaran->delete();
        //paket with this pelajaran delete
        return response()->json($request-
>id);
    }

```

```

        public function deleteAll(Request
$request)
        {
            $pakets =
            Paket::where('pelajaran_id', $request->id)-
            >get();
            foreach ($pakets as $paket) {
                $paket->jadwalujians()-
            >detach();
                $soals = Soal::where('paket_id',
                $paket->id)->get();
                foreach ($soals as $soal) {
                    $soal->delete();
                }
                $ujian_siswas =
            UjianSiswa::where('paket_id', $paket->id)-
            >get();
                foreach ($ujian_siswas as
                $ujian_siswa) {
                    $ujian_siswa->delete();
                }
                $paket->delete();
            }
            return response()->json();
        }
    }

```

13. Peserta Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Uuid;

use App\Models\User;
use App\Models\Sekolah;
use App\Models\JadwalUjian;
use App\Models\UjianSiswa;

```

```

class PesertaController extends Controller
{
    public function index()
    {
        $jadwals = JadwalUjian::all();
        $options = '';
        foreach ($jadwals as $jadwal)
        {
            if($jadwal->ujianSiswas-
>count() == 0)
                $options .= '<option
value="'. $jadwal->id.'">Ujian:      ' . $jadwal-
>deskripsi.'          (sesi:      ' . $jadwal-
>sesi.)</option>';
        }
        return
view('pages.proktor.sesi.sesi',
compact('options'));
    }

    public function showDatatables()
    {
        return
view('pages.proktor.sesi.datatables');
    }

    public function masukkanKeUjian()
    {
        $sesi_ids =
array_unique(request('sesi'));
        $siswas = User::whereIn('id',
request('siswas'))->get();
        foreach ($sesi_ids as $id_jadwal)
        {
            $jadwal =
JadwalUjian::find($id_jadwal);
            foreach ($siswas as $siswa)
            {

```

```

        $data['user_id'] = $siswa-
>id;
        $data['id'] =
Uuid::generate()->string;
        $data['paket_id'] = "0";
        $data['jadwal_ujian_id'] =
$jadwal->id;
        $data['waktu_mulai'] =
$jadwal->waktu_mulai;
        $data['waktu_selesai'] =
$jadwal->waktu_selesai;
        $ujian_siswa =
UjianSiswa::create($data);
    }
    $this->random($jadwal->id);
}

return redirect('/jadwal')-
>with('success', 'Peserta ujian berhasil
diperbaharui');
}

public function random($jadwal_id){
    $jadwal_ujian =
JadwalUjian::find($jadwal_id);
    $pakets = $jadwal_ujian->pakets;
    $ujian_siswas = $jadwal_ujian-
>ujianSiswas;
    foreach($ujian_siswas as
$ujian_siswa){
        // dd($ujian_siswa);
        $i = rand(0,$pakets->count()-1);
        $paket = $pakets[$i];
        $arr_soal_id_with_essai =
array();

        //pilgan
        $soals = $paket->soals-
>where('tipe_soal','pilihan_ganda');

```

```

        $arr_soal_id = array();
        $arr_random_jawaban = array();
        $arr_jawaban
    array("1", "2", "3", "4", "5");
        $arr_jawaban_siswa = array();
        foreach($soals as $soal){
            shuffle($arr_jawaban);

array_push($arr_soal_id, $soal->id);

array_push($arr_random_jawaban, $arr_jawaban
);

array_push($arr_jawaban_siswa, " ");
        }
        shuffle($arr_soal_id);
        // dd($arr_soal_id);
        foreach ($arr_soal_id as
$soal_id) {

array_push($arr_soal_id_with_essai,
$soal_id);
        }
        // dd($arr_soal_id_with_essai);

        //essai
        $soals_essai = $paket->soals-
>where('tipe_soal', 'essai');
        foreach ($soals_essai as
$soal_essai) {

array_push($arr_soal_id_with_essai,
$soal_essai->id);
        }
        // dd($arr_soal_id_with_essai);

        $ujian_siswa->paket_id = $paket-
>id;

```



```

        $ujian_siswa->random_soal      =
json_encode($arr_soal_id_with_essai);
        $ujian_siswa->random_jawaban  =
json_encode($arr_random_jawaban);
        $ujian_siswa->jawaban_siswa   =
json_encode($arr_jawaban_siswa);
        $ujian_siswa->save();
    }
    // dd($ujian_siswas[0]);
    return;
}

public function datatables(Request
$request)
{
    //Isi dengan kolom
    $columns = array(
        0 => 'id',
        1 => 'nisn',
        2 => 'nama',
        3 =>
'jenis_kelamin',
        4 => 'sekolah_id'
    );

    //<-- Gak Perlu Diubah -->
    $totalData =
count(User::where('level', 'siswa')->get());

    $totalFiltered = $totalData;

    $limit = $request->input('length');
    $start = $request->input('start');
    $order = $columns[$request-
>input('order.0.column')];
    $dir = $request-
>input('order.0.dir');

```

```

        if(empty($request-
>input('search.value'))
        {
            $siswas =
User::where('level','siswa')
                                ->offset($start)
                                ->limit($limit)
                                -
>orderBy($order,$dir)
                                ->get();
        }
        else {
            $search = $request-
>input('search.value');
            //<-- Gak Perlu Diubah END -->
            //Tambah orWhere di kolom yang akan
            dijadikan di search
            $siswas = User::where('level',
'siswa')
                                -
>where(function ($q) use ($search) {
                                $q-
>where('nama','LIKE','"%{$search}%"')
                                -
>orWhere('nisn', 'LIKE','"%{$search}%"')
                                -
>orWhereHas('sekolah', function($query) use
($search) {
                                $query-
>where('nama', 'LIKE', "%{$search}%"');
                                });
                                });
                                -
>offset($start)
                                ->limit($limit)
                                -
>orderBy($order,$dir)
                                ->get();

```

```

        //Tambah orWhere di kolom yang akan
        dijadikan di search
        $totalFiltered =
User::where('level', 'siswa')
-
>where(function ($q) use ($search) {
        $q-
>where('nama', 'LIKE', "%{$search}%")
-
>orWhere('nisn', 'LIKE', "%{$search}%")
-
>orWhereHas('sekolah', function($query) use
($search){
        $query-
>where('nama', 'LIKE', "%{$search}%");
        });
    })
    ->count();
}

    $data = array();
    $data_siswas = $siswas;
    if(!empty($data_siswas))
    {
        foreach ($data_siswas as $index
=> $siswa)
        {
            //Apa yang akan ditampilkan
            di tiap2 row
            $nestedData[0] = $siswa->id;
            $nestedData[1] = $siswa->
>nisn;
            $nestedData[2] = $siswa->
>nama;
            $nestedData[3] = $siswa->
>jenis_kelamin;
            $sekolah =
Sekolah::find($siswa->sekolah_id);
            if($siswa->sekolah_id)

```

```

        {
            $sekolah = $siswa->sekolah->nama;
        }
        else
        {
            $sekolah = 'Data Sekolah Siswa Belum Terinput';
        }
        $nestedData[4] = $sekolah;

        $data[] = $nestedData;
    }
}

//<-- Gak Perlu Diubah -->
$json_data = array(
    "draw" =>
intval($request->input('draw')),
    "recordsTotal" =>
intval($totalData),
    "recordsFiltered" =>
intval($totalFiltered),
    "data" =>
$data
);

return json_encode($json_data);
//<-- Gak Perlu Diubah END -->
}
}

```

14. Pilihan Ganda Controller

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Symfony\Component\Process\Process;
use
Symfony\Component\Process\Exception\Process
FailedException;
use Exception;
use Input;
use URL;
use App\Models\UjianSiswa;
use App\Models\Sekolah;
use App\Models\JadwalUjian;
use App\Models\Soal;
use App\Models\JawabanEssai;

class PilganController extends Controller
{
    public function pilihSoal($sekolah_id,
$id)
    {
        $data['id'] = $id;
        $data['ujian'] =
JadwalUjian::find($id);
        $data['sekolah'] =
Sekolah::find($sekolah_id);
        $data['ujians_sekolah'] =
$data['sekolah']->ujians()-
>where('jadwal_ujian_id', $id)-
>paginate(10);
        return
view('pages.admin.korektor.pilih_siswa',
$data);
    }

    public function pilihSekolah($id) {
        $data['sekolahs'] =
Sekolah::where('id', 'not like', '%hide%')-
>get();
    }
}

```

```

        $data['id'] = $id;
        return
    view('pages.admin.korektor.pilih_sekolah',
        $data);
    }

    public function
    detailJawaban($jadwal_id, $user_id)
    {
        $url = url()->previous();
        $data['ujian_siswa'] =
        UjianSiswa::where('jadwal_ujian_id',
        $jadwal_id)->where('user_id', $user_id)-
        >first();
        $soals =
        json_decode($data['ujian_siswa']-
        >random_soal);
        $jawabans_siswa =
        json_decode($data['ujian_siswa']-
        >jawaban_siswa);
        $jawaban_posisi =
        json_decode($data['ujian_siswa']-
        >random_jawaban);

        $jumlah = count($jawabans_siswa);
        $data['results'] = array();
        $data['arr_random'] = array();
        $data['new_jawabans'] = array();
        for ($i=0; $i < $jumlah; $i++) {
            $inttochar = array("1" =>
            "a", "2"=>"b", "3"=>"c", "4"=>"d", "5"=>"e");
            $data['arr_random'] = array("a"
            => $inttochar[$jawaban_posisi[$i][0]],
            "b" =>
            $inttochar[$jawaban_posisi[$i][1]],
            "c" =>
            $inttochar[$jawaban_posisi[$i][2]],
            "d" =>
            $inttochar[$jawaban_posisi[$i][3]],

```

```

                                "e" =>
$inttochar[$jawaban_posisi[$i][4]]);
        if($jawabans_siswa[$i]== " ")
            $data['new_jawabans'][$i] =
'kosong';
        else{
            $data['new_jawabans'][$i] =
$data['arr_random'][strtolower($jawabans_si
swa[$i])];
        }
        $hehe['kunci'][$i] =
$data['ujian_siswa']->soal($soals[$i]) -
>kunci_jawaban;
        $data['results'][$i] =
array($soals[$i],
$data['new_jawabans'][$i]);
    }
    $uri = explode('/', $url);
    $data['url_prev'] =
$uri[3].'/'. $uri[4];
    return
view('pages.admin.korektor.pilgan.index',
$data);
}

    public function koreksiUlang(Request
$request)
    {
        $jadwal_ujian =
JadwalUjian::find($request->ujian_id);
        $sekolah = Sekolah::where('nama',
$request->nama_sekolah)->first();
        if($sekolah)
        {
            $ujian_siswas = $sekolah->
ujians()->where('jadwal_ujian_id',
$request->ujian_id)->get();
            foreach($ujian_siswas as
$ujian_siswa){

```

```

        try{
            $soals =
json_decode($ujian_siswa->random_soal);
            $jawaban_posisi =
json_decode($ujian_siswa->random_jawaban);
            $jawabans =
(array)json_decode($ujian_siswa-
>jawaban_siswa);

            $jawaban_benar = 0;
            $jawaban_kosong = 0;
            $jawaban_salah = 0;
            foreach($soals as $key
=> $soal_id){
                $soal =
Soal::find($soal_id);
                if($soal->tipe_soal
== "pilihan_ganda"){

                    if(!isset($jawabans[$key])){
                        throw new
Exception("Notice: Undefined offset:
".$key);
                    }

                    if($jawabans[$key] == " "){
                        $jawaban_kosong+=1;
                    }else{
                        $inttochar =
array("1" =>
"a", "2"=>"b", "3"=>"c", "4"=>"d", "5"=>"e");
                        $arr_random =
array("a" =>
$inttochar[$jawaban_posisi[$key][0]],
                        "b" => $inttochar[$jawaban_posisi[$key][1]],
                        "c" => $inttochar[$jawaban_posisi[$key][2]],

```



```

"d" => $inttochar[$jawaban_posisi[$key][3]],
"e" =>
$inttochar[$jawaban_posisi[$key][4]]);

if($arr_random[strtolower($jawabans[$key])]
== strtolower($soal->kunci_jawaban)){

$jawaban_benar+=1;
} else{

$jawaban_salah+=1;
}
}
}
else if ($soal-
>tipe_soal == 'esai')
{
    $jawaban_essai =
JawabanEssai::where('ujian_siswa_id',
$ujian_siswa->id)->where('soal_id', $soal-
>id)->first();

if(!$jawaban_essai || !$jawaban_essai-
>jawaban_siswa)

$jawaban_kosong+=1;
}

    $ujian_siswa-
>jumlah_benar = $jawaban_benar;
    $ujian_siswa-
>jumlah_salah = $jawaban_salah;
    $ujian_siswa-
>jumlah_kosong = $jawaban_kosong;
    $ujian_siswa-
>save();
}

```

```

        } catch (Exception $e) {
            $ujian_siswa-
>jumlah_benar = -2;
            $ujian_siswa-
>jumlah_salah = -2;
            $ujian_siswa-
>jumlah_kosong = -2;
            $ujian_siswa->save();
        }
        $jadwal_ujian->status =
'Terkoreksi';
        $jadwal_ujian->save();
        return redirect()->back();
    }
    return redirect()->back();
}
}

```

15. Proktor Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProktorController extends Controller
{
    public function index()
    {
        return
view('pages.proktor.ganti_password');
    }

    public function changePassword()
    {
        $this->validate(request(), [
            'password'
            =>
'min:6|required|confirmed'

```

```

    });

    $user = auth()->user();
    $user->password =
bcrypt(request('password'));
    $user->save();

    return back()->with('success',
'Password berhasil diubah!');
    }
}

```

16. Sekolah Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Uuid;
use App\Models\Sekolah;

class SekolahController extends Controller
{
    /* Constructor
    -Only auth
    */
    public function __construct()
    {
        // $this->middleware('auth');
    }

    /* Read
    - Read
    tampilkan sekolah
    */
    public function index($id)
    {

```

```

        $kode_kota = '___'.$id.'%';
        $sekolahs
Sekolah::where('kode','LIKE',$kode_kota)-
>orderBy('created_at','asc')->get();
        $no = 1;
        return
view('pages/admin/sekolah/sekolah',
compact('sekolahs','no'));
    }

    /* Edit
    - Edit
        tampilan editing nya pake modal
        berupa ajax request
    */
    public function edit()
    {
        $data = request()->all();

        $sekolah
Sekolah::find($data['sekolah_id']);
        $sekolah->nama = $data['nama'];
        $sekolah->alamat = $data['alamat'];
        $sekolah->kurikulum
        $data['kurikulum'];
        $sekolah->npsn = $data['npsn'];
        $sekolah->kode = $data['kode'];
        $sekolah->save();

        return redirect()-
>route('sekolah.kota','sekolah')-
>with('flash_success','Sekolah berhasil
diperbarui');
    }

    public function create()
    {
        $data = request()->all();

```

```

        $data['id'] = Uuid::generate()-
>string;
        $sekolah = Sekolah::create($data,
['except' => ['_token', 'submit']]);

        return redirect()-
>route('sekolah.kota','sekolah')-
>with('flash_success', 'Sekolah berhasil
dibuat');
    }

    public function delete()
    {
        $id = request('id');
        $sekolah = Sekolah::find($id);
        $sekolah->delete();
        // $servers =
Server::where('sekolah_id',$id);
        // $servers->delete();
        return response()->json($id);
    }

    public function getSekolah()
    {
        $id = request('id');
        $sekolah = Sekolah::find($id);

        return response()->json($sekolah);
    }

    public function getSchools(Request
$request)
    {
        $term = trim($request->q);
        if (empty($term)) {
            return response()->json([]);
        }
        $schools = Sekolah::where('nama',
'LIKE', "%{$term}%")->limit(5)

```

```

        ->get ();

        $formatted_tags = [];
        foreach ($schools as $tag) {
            $formatted_tags[] = ['id' =>
$tag->id, 'text' => $tag->nama, 'value' =>
$tag->id];
        }
        return response()-
>json($formatted_tags);
    }
}

```

17. Serial Number Controller

```

<?php

namespace App\Http\Controllers;

use App\Models\Sekolah;
use App\Models\SerialNumber;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Session;
use
Symfony\Component\Process\Exception\Process
FailedException;
use Symfony\Component\Process\Process;

class SerialNumberController extends
Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
}

```

```

        */
        public function index()
        {
            //
        }

        /**
         * Show the form for creating a new
        resource.
         *
         * @return \Illuminate\Http\Response
         */
        public function create()
        {
            //
        }

        /**
         * Store a newly created resource in
        storage.
         *
         * @param \Illuminate\Http\Request
        $request
         * @return \Illuminate\Http\Response
         */
        public function store(Request $request)
        {
            //
        }

        /**
         * Display the specified resource.
         *
         * @param \App\Models\SerialNumber
        $serialNumber
         * @return \Illuminate\Http\Response
         */
        public function show(SerialNumber
        $serialNumber)

```

```

    {
        //
    }

    /**
     * Show the form for editing the
    specified resource.
     *
     * @param \App\Models\SerialNumber
    $serialNumber
     * @return \Illuminate\Http\Response
     */
    public function edit(SerialNumber
    $serialNumber)
    {
        //
    }

    /**
     * Update the specified resource in
    storage.
     *
     * @param \Illuminate\Http\Request
    $request
     * @param \App\Models\SerialNumber
    $serialNumber
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request,
    SerialNumber $serialNumber)
    {
        //
    }

    /**
     * Remove the specified resource from
    storage.
     *

```



```

        * @param \App\Models\SerialNumber
$serialNumber
        * @return \Illuminate\Http\Response
        */
        public function destroy(SerialNumber
$serialNumber)
        {
            //
        }

        public function register(Request
$request){
            $server_id =
SerialNumber::where('server_id',$request-
>server_id)->first();
            if (!$server_id) {
                return response()->json('server
id tidak ditemukan',404);
            }

            $server_id->sn = $request->SN;
            $server_id->state = 1;

            $sekolah_id = $server_id-
>sekolah_id;

            $sekolah_nama =
Sekolah::find($sekolah_id)->nama;

            $server_id->save();
            return response()-
>json(['sekolah_id' =>
$sekolah_id,'sekolah_nama' =>
$sekolah_nama]);
        }

        public function check(Request $request){
            $version = $request->version;
            $process = null;

```

```

        try {
            $process = new Process('git
describe origin/dev-jatim-bawah-2020 --
tag');

            $process->run();

            if (!$process->isSuccessful()) {
                throw new
ProcessFailedException($process);
            }
        } catch (\Throwable $th) {
            return response()-
>json('old_version',400);
        }
        $recent_ver = explode("\n",
$process->getOutput());
        $recent_ver =
$recent_ver[sizeof($recent_ver)-2];
        $recent_ver = explode("-
",$recent_ver);
        $recent_ver = $recent_ver[0].'-
'.$recent_ver[1];

        $version = explode("-", $version);
        $version = $version[0].'-
'.$version[1];

        if ($version != $recent_ver) {
            return response()-
>json('old_version',400);
        }

        $sn = $request->sn;
        $server_id = $request->server_id;
        $server =
=
        Serializable::where('server_id',$server_id)
->first();
        if (!$server) {

```

```

        return response() -
>json('not_found',400);
    }

    if (!$server->sn) {
        $server->sn = $sn;
    }

    else if ($server->sn != $sn) {
        return response() -
>json('no_match',400);
    }

    // dd($server,$sn, $server_id);
    // var_dump($sn);
    // die();
    try {
        $server->state = 1;
        $server->save();
    } catch (\Throwable $th) {
        return response() -
>json('used_sn',400);
    }

    $sekolah_id = $server->sekolah_id;

    $sekolah_nama =
Sekolah::find($sekolah_id)->nama;

    return response() -
>json(['sekolah_id' =>
$sekolah_id,'sekolah_nama' =>
$sekolah_nama],200);
    }

    public function requestServer()
    {

```

```

        $serial_number =
        SerialNumber::where('sekolah_id',Auth::user
        ()->sekolah_id)->get();
        return
        view('pages/proktor/serial_number/index',
        compact('serial_number'));
    }

    public function
    requestServerSubmit(Request $request)
    {
        return redirect()->back()-
        >with('error','Request server dikunci!');
        $sekolah =
        Sekolah::find(Auth::user()->sekolah_id);
        $jumlah_sn =
        SerialNumber::where('sekolah_id','=',$sekol
        ah->id)->count();
        if ($jumlah_sn + $request-
        >jumlah_server > 25) {
            return redirect()->back()-
            >with('error', 'Jumlah Request Server Telah
            Mencapai Batas. Maksimal 25 Server!');
        }
        if ($request->jumlah_server < 1) {
            return redirect()->back()-
            >with('error', 'Format Salah!');
        }
        for ($i=0; $i < $request-
        >jumlah_server; $i++, $jumlah_sn++) {
            $serial_number = new
            SerialNumber();
            $serial_number->sekolah_id =
            $sekolah->id;
            $serial_number->sn = null;
            $serial_number->server_id =
            'U'.$sekolah->kode.'-
            '.$jumlah_sn.strtoupper(str_random(3));
            $serial_number->state = 0;

```

```

        $serial_number->save();
    }
    return redirect()->back()-
>with('success', 'Jumlah Server Berhasil
Ditambahkan');
}

public function showMonitoring()
{
    return
view('pages.admin.serialnumber.index');
}

public function indexSekolah()
{
    $sekolahs =
Sekolah::orderBy('created_at', 'asc')-
>get();
    $no = 1;
    return
view('pages.admin.serialnumber.indexsekolah
', compact('sekolahs','no'));
}

public function deleteSN()
{
    $id = request('id');
    $serialnumber =
SerialNumber::where('sekolah_id',$id);
    $serialnumber->delete();
    return response()->json($id);
}

public function getJmlServer()
{
    $id = request('id');
    $sekolah = Sekolah::find($id);

```

```

        $serialnumber =
        SerialNumber::where('sekolah_id',$id)-
        >count();
        $sekolah->jmlServer = $serialnumber;
        return response()->json($sekolah);
    }

    public function editJmlServer(Request
$request)
    {
        $sekolah = Sekolah::find($request-
        >sekolah_id);
        $jumlah_sn =
        SerialNumber::where('sekolah_id','=', $sekol
        ah->id)->count();
        $sekolah_nama = explode('
',$sekolah);
        $server_id =
        $sekolah_nama[1].substr($sekolah_nama[2],0,
        3);
        for ($i=0; $i < $request-
        >tmbhServer; $i++) {
            $serial_number = new
            SerialNumber();
            $serial_number->sekolah_id =
            $sekolah->id;
            $serial_number->sn = null;
            $serial_number->server_id =
            'U'.$sekolah->kode.'-
            '.$jumlah_sn.strtoupper(str_random(3));
            $serial_number->state = 0;
            $serial_number->save();
        }
        return redirect()->back()-
        >with('success', 'Jumlah Server Berhasil
        Ditambahkan');
    }

    public function showSN($id)

```

```

        {
            $sekolah = Sekolah::find($id);
            $serial_number =
SerialNumber::where('sekolah_id',$id)-
>get();
            return
view('pages.admin.serialnumber.detail',
compact('sekolah','serial_number'));
        }

        public function resetSN(Request
$request)
        {
            $serial_number =
SerialNumber::where('server_id',$request-
>server_id)->first();
            $serial_number->sn = NULL;
            $serial_number->save();

            $server_id = $request->server_id;
            $this->logAdmin('Reset
SN',$server_id);
            return redirect()->back()-
>with('success','SN berhasil direset!');
        }

        public function beriAkses(Request
$request) {
            $serial_number =
SerialNumber::where('server_id',$request-
>server_id)->first();
            $serial_number->permission = 1;
            $serial_number->save();

            $server_id = $request->server_id;
            $this->logAdmin('Beri akses rubah
jadwal',$server_id);

```

```

        return redirect()->back()-
>with('success','Hak izin berhasil
diberikan!');
    }

    public function cabutAkses(Request
$request) {
        $serial_number =
        SerialNumber::where('server_id',$request-
>server_id)->first();
        $serial_number->permission = 0;
        $serial_number->save();

        $server_id = $request->server_id;
        $this->logAdmin('Cabut akses rubah
jadwal',$server_id);
        return redirect()->back()-
>with('success','Hak izin berhasil
dicabut!');
    }

    public function showLog(){

        $log = DB::table('log_sn')->get();
        return
        view('pages.admin.serialnumber.actionlog',
        compact('log'));
    }

    public function ajaxLog(Request
$request){
        //Isi dengan kolom

        $columns = array(
            0 =>'nama_user',
            1 =>'penjelasan',
            2 =>'server_id',
            3 =>'created_at',
        );
    }

```



```

        //<-- Gak Perlu Diubah -->
        $totalData = DB::table('log_sn')-
>count();

        $totalFiltered = $totalData;

        $limit = $request->input('length');
        $start = $request->input('start');
        $order = $columns[$request-
>input('order.0.column')];
        $dir = $request-
>input('order.0.dir');

        if(empty($request-
>input('search.value')))
        {
            $logs = DB::table('log_sn')-
>offset($start)
            ->limit($limit)
            ->orderBy($order,$dir)
            ->get();
            // return response() -
>json('here',200);
        }
        else {
            $search = $request-
>input('search.value');
            //<-- Gak Perlu Diubah END -->
            //Tambah orWhere di kolom yang akan
            dijadikan di search
            $logs = DB::table('log_sn')-
>where('nama_user','LIKE','"%{$search}%")
            // -
>orWhere('title','LIKE','"%{$search}%")
            -
>offset($start)
            ->limit($limit)

```

```

-
>orderBy($order,$dir)
                                ->get();
    //Tambah orWhere di kolom yang akan
    dijadikan di search
    $totalFiltered =
DB::table('log_sn')-
>where('nama_user','LIKE','"%{$search}%"')
                                //
>orWhere('title','LIKE','"%{$search}%"')
                                ->count();
    }

    $data = array();
    if(!empty($logs))
    {
        foreach ($logs as $log)
        {
            //Apa yang akan ditampilkan
            di tiap2 row
            $nestedData['nama_user'] =
$log->nama_user;
            $nestedData['penjelasan'] =
$log->penjelasan;
            $nestedData['server_id'] =
$log->server_id;
            $nestedData['created_at'] =
$log->created_at;

            $data[] = $nestedData;
        }
    }

    //<-- Gak Perlu Diubah -->
    $json_data = array(
        "draw" =>
intval($request->input('draw')),
        "recordsTotal" =>
intval($totalData),

```

```

                                "recordsFiltered" =>
intval($totalFiltered),
                                "data" =>
$data
                                );

        return json_encode($json_data);
        //<-- Gak Perlu Diubah END -->

    }

    public function deleteServer(Request
$request) {
        $server_id = $request->server_id;
        $server =
        =
        SerialNumber::where('server_id',$server_id)
->first();
        $sekolah = $server->sekolah;
        if (Auth::user()->sekolah_id !=
        $sekolah->id) {
            return redirect()->back()-
            with('error','Terjadi error dalam menghapus
            id server');
        }
        $server->delete();
        return redirect()->back()-
        >with('success','id server berhasil
        dihapus');
    }

    public function adminDelete(Request
$request) {
        $server_id = $request->server_id;
        $server =
        =
        SerialNumber::where('server_id',$server_id)
->first();
        $server->delete();
    }

```

```

        $this->logAdmin('Hapus
server',$server_id);
        return redirect()->back()-
>with('success','id server berhasil
dihapus');
    }

    private function
logAdmin($penjelasan,$server_id) {

        $nama = Auth::user()->nama;
        if (!$nama) {
            $nama = Auth::user()->id;
        }

        $query = array([
            'server_id' => $server_id,
            'nama_user' => $nama,
            'penjelasan' => $penjelasan,
            'created_at' => Carbon::now()
        ]);

        $db = DB::table('log_sn')-
>insert($query);
        return 'sukses';

    }

    public function ajaxDatatables(Request
$request) {
        //Isi dengan kolom
        $columns = array(
            0 => 'server_id',
            1 => 'sekolah_nama',
            2 => 'sn',
            3 => 'permission',
            4 => 'sn_reset',
            5 => 'action',
        );
    }

```

```

    //<-- Gak Perlu Diubah -->
    $totalData = SerialNumber::count();

    $totalFiltered = $totalData;

    $limit = $request->input('length');
    $start = $request->input('start');
    $order = $columns[$request-
>input('order.0.column')];
    $dir = $request-
>input('order.0.dir');
    $check_arr =
['sn', 'server_id', 'sn_reset', 'permission'];
    if (!in_array($order, $check_arr)) {
        return response()-
>json('Error', 400);
    }
    if(empty($request-
>input('search.value')))
    {
        $q = "SELECT sn.* , lsn.sn_reset
FROM serial_numbers sn LEFT JOIN (SELECT
server_id, COUNT(*) as sn_reset FROM log_sn
WHERE penjelasan = 'Reset SN' GROUP BY
server_id) lsn ON sn.server_id =
lsn.server_id";
        if ($dir == 'asc') {
            $q = $q." ORDER BY ".$order."
ASC";
        } else {
            $q = $q." ORDER BY ".$order."
DESC";
        }
        $q = $q." LIMIT :limit";
        $q = $q." OFFSET :start";
        // DB::enableQueryLog();
    }

```

```

        $serial_numbers =
DB::select($q,[
        'limit' => $limit,
        'start' => $start
    ]);
    }
    else {
        $search = $request->
input('search.value');

        //<-- Gak Perlu Diubah END -->
        $q = "SELECT sn.* , lsn.sn_reset
FROM serial_numbers sn LEFT JOIN (SELECT
server_id, COUNT(*) as sn_reset FROM log_sn
WHERE penjelasan = 'Reset SN' GROUP BY
server_id) lsn ON sn.server_id =
lsn.server_id where";
        $q = $q." sn.server_id
like :search1";
        $q = $q." OR sn.sn
like :search2";
        if ($dir == 'asc') {
            $q = $q." ORDER BY ".$order."
ASC";
        } else {
            $q = $q." ORDER BY ".$order."
DESC";
        }
        $q = $q." LIMIT :limit";
        $q = $q." OFFSET :start";
        // DB::enableQueryLog();

        $serial_numbers =
DB::select($q,[
        'search1' =>
        '%'.$search.'%',
        'search2' =>
        '%'.$search.'%',
        'limit' => $limit,

```

```

        'start' => $start
    ]);

    //Tambah orWhere di kolom yang akan
    dijadikan di search
    $totalFiltered =
count($serial_numbers);
    }

    $data = array();
    if(!empty($serial_numbers))
    {
        foreach ($serial_numbers as
$serial_number)
        {
            //Apa yang akan ditampilkan
            di tiap2 row
            $nestedData['server_id'] =
$serial_number->server_id;
            // dd($serial_number-
>sekolah);
            $sekolah =
Sekolah::find($serial_number->sekolah_id);
            $nestedData['sekolah_nama']
= $sekolah->nama;
            $nestedData['sn'] =
$serial_number->sn;
            $nestedData['permission'] =
$serial_number->permission;
            if ($serial_number-
>sn_reset) {
                $jumlah_sn =
$serial_number->sn_reset;
            } else {
                $jumlah_sn = 0;
            }
            $nestedData['sn_reset'] =
$jumlah_sn;
            $nestedData['action'] =

```

```

'
        <td>
            <form
action="/serialnumber/reset_sn"
method="post">
                <input type="text"
name="_token" value="''.Session::token().'"
hidden>
                <input type="text"
name="server_id" value="''. $serial_number-
>server_id.'" hidden>
                <button class="btn
btn-primay" type="submit">Reset SN</button>
            </form>
        </td>
        <td>
            <form
action="/serialnumber/server/delete"
method="post">
                <input type="text"
name="_token" value="''.Session::token().'"
hidden>
                <input type="text"
name="server_id" value="''. $serial_number-
>server_id.'" hidden>
                <button class="btn
btn-primay deleteButton" type="button"
onClick=deleteButton(this)>Delete
ID</button>
            </form>
        </td>>;
        if($serial_number-
>permission){
            $nestedData['action'] =
$nestedData['action']. '
            <form
action="/serialnumber/revoke_permission"
method="post">

```



```

name="_token"    value="'.Session::token().'"
hidden>
                                <input    type="text"
name="server_id"    value="'. $serial_number-
>server_id.'" hidden>
                                <button    class="btn
btn-primay"        type="submit">Cabut
Izin</button>
                                </form>';
                                } else {
                                $nestedData['action'] =
$nestedData['action']. '
                                <form
action="/serialnumber/give_permission"
method="post">
                                <input    type="text"
name="_token"    value="'.Session::token().'"
hidden>
                                <input    type="text"
name="server_id"    value="'. $serial_number-
>server_id.'" hidden>
                                <button    class="btn
btn-primay"        type="submit">Izinkan    Ubah
Jadwal</button>
                                </form>';
                                }
                                $data[] = $nestedData;
                                }
                                }

                                //<-- Gak Perlu Diubah -->
                                $json_data = array(
                                    "draw" =>
intval($request->input('draw')),
                                    "recordsTotal" =>
intval($totalData),
                                    "recordsFiltered" =>
intval($totalFiltered),

```

```

                                "data"                                =>
$data
                                );

                                return json_encode($json_data);
                                //<-- Gak Perlu Diubah END -->
                                }
                                }

```

18. Server Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Uuid;
use App\Models\Sekolah;
use App\Models\Server;

class ServerController extends Controller
{
    /* Constructor
       -Only auth
    */
    public function __construct()
    {
        // $this->middleware('auth');
    }

    /* Read
       - Read
       tampilan sekolah
    */
    public function index($sekolah_id)
    {
        $sekolah =
        Sekolah::find($sekolah_id);
        if (!$sekolah) return
        redirect('sekolah')->withErrors(['

```

```

        message' => 'Sekolah
        '.$sekolah_id.' belum dibuat'
    });

    $servers =
Server::where('sekolah_id', $sekolah_id)-
>orderBy('created_at', 'asc')->get();

    $no = 1;
    return
view('pages/admin/server/server',
compact('servers','sekolah','no'));
}

/* Edit
- Edit
    tampilan editing nya pake modal
    berupa ajax request
*/
public function edit()
{
    $data = request()->all();

    $server =
Server::find($data['server_id']);

    $server->nama = $data['nama'];
    $server->ip = $data['ip'];

    $server->save();

    return back();
}

public function create()
{
    $data = request()->all();
    $data['id'] = Uuid::generate()-
>string;

```

```

        // dd($data);
        $server = Server::create($data,
['except' => ['_token', 'submit']]);

        return back();
    }

    public function delete()
    {
        $id = request('id');
        $server = Server::find($id);
        $server->delete();

        return response()->json($id);
    }

    public function getServer()
    {
        $id = request('id');
        $server = Server::find($id);

        return response()->json($server);
    }
}

```

19. Siswa Controller

```

<?php

namespace App\Http\Controllers;

use Uuid;
use Carbon\Carbon;
use Zipper;
use Illuminate\Validation\Rule;
use Illuminate\Support\Facades\Validator;
use App\Models\User;
use App\Models\Sekolah;

```

```

use App\Models\UjianSiswa;
use App\Models\Server;
use App\Models\JawabanEssai;
use App\Models\JadwalUjian;
use App\Models\KabKota;
use Illuminate\Http\Request;
use Maatwebsite\Excel\Facades\Excel;
use Illuminate\Support\Facades\Storage;
use PDF;

class SiswaController extends Controller
{
    /* Constructor
       -Only auth
    */
    public function __construct()
    {
        // $this->middleware('auth');
    }

    /* Read
       - Read
       tampilkan soal menurut paket
    */

    public function kota($id)
    {
        $data['id'] = $id;
        $data['kota'] = KabKota::orderBy('kd_rayon')->get();
        $data['no'] = 1;
        return view('pages.admin.siswa.kota', $data);
    }

    public function sekolah($id)
    {
        $kode_kota = ' ' . $id . '%';
    }
}

```

```

        $data['kota'] =
KabKota::where('kd_rayon', $id)->first();
        $data['sekolah'] =
Sekolah::where('kode', 'LIKE', $kode_kota)-
>get();
        $data['no'] = 1;
        return
view('pages.admin.siswa.sekolah', $data);
    }

    public function all($id_sekolah)
    {
        $data['sekolah'] =
Sekolah::where('id', $id_sekolah)->first();
        $data['siswa'] =
User::where('sekolah_id', $id_sekolah)-
>where('level', 'siswa')->get();
        $data['no'] = 1;
        return view('pages.admin.siswa.all',
$data);
    }

    public function
listUjianSiswa($id_siswa)
    {
        $data['ujian_siswas'] =
UjianSiswa::where('user_id', $id_siswa)-
>get();
        $data['user_id'] = $id_siswa;
        return
view('pages.admin.user.ujian', $data);
    }

    public function
averageUjianSiswa($id_siswa, $ujian_id)
    {
        $siswa = User::find($id_siswa);
        $jadwal =
JadwalUjian::find($ujian_id);

```

```

        $sekolah = Sekolah::find($siswa-
>sekolah_id);

        $pelajaran = $jadwal->pelajaran;
        $jadwals = $pelajaran->jadwals-
>pluck('id');
        $data['ujians'] = $sekolah->ujians-
>whereIn('jadwal_ujian_id', $jadwals)-
>where('jumlah_benar', '>', 0);
        return
view('pages.admin.user.average', $data);
    }
    public function updateUjianSiswa(Request
$request)
    {
        $ujian_siswa =
UjianSiswa::find($request->ujian_siswa_id);
        $counter = $request->jumlah_benar-
$ujian_siswa->jumlah benar;
        if($counter<0) {
            return redirect()->back();
        }
        $cek = 0;
        return $ujian_siswa->kunci_jawaban;
        foreach ($ujian_siswa->kunci_jawaban
as $key => $kunci) {
            $cek++;
        }
        return $cek;
    }
    public function index()
    {
        return
view('pages.proktor.siswa.index');
    }

    public function datatables(Request
$request)
    {

```

```

        //Isi dengan kolom
        $columns = array(
            0 => 'nisn',
            1 => 'nama',
            2 =>
'jenis_kelamin',
            3 => 'sekolah_id',
            4 => 'username',
            5 => 'action'
        );

        //<-- Gak Perlu Diubah -->
        $totalData =
count(User::where('level', 'siswa')->get());

        $totalFiltered = $totalData;

        $limit = $request->input('length');
        $start = $request->input('start');
        $order = $columns[$request-
>input('order.0.column')];
        $dir = $request-
>input('order.0.dir');

        if(empty($request-
>input('search.value')))
        {
            $siswas =
User::where('level', 'siswa')
                ->offset($start)
                ->limit($limit)
                -
>orderBy($order, $dir)
                ->get();
        }
        else {
            $search = $request-
>input('search.value');
            //<-- Gak Perlu Diubah END -->

```



```

        //Tambah orWhere di kolom yang akan
        dijadikan di search
        $siswas = User::where('level',
'siswa')
-
>where(function ($q) use ($search) {
        $q-
>where('nama', 'LIKE', "%{$search}%")
-
>orWhere('nisn', 'LIKE', "%{$search}%")
-
>orWhere('username', 'LIKE', "%{$search}%")
-
>orWhereHas('sekolah', function($query) use
($search){
        $query-
>where('nama', 'LIKE', "%{$search}%");
        });
-
>offset($start)
-
->limit($limit)
-
>orderBy($order,$dir)
-
        ->get();
        //Tambah orWhere di kolom yang akan
        dijadikan di search
        $totalFiltered =
User::where('level', 'siswa')
-
>where(function ($q) use ($search) {
        $q-
>where('nama', 'LIKE', "%{$search}%")
-
>orWhere('nisn', 'LIKE', "%{$search}%")
-
>orWhere('username', 'LIKE', "%{$search}%")

```

```

-
>orWhereHas('sekolah', function($query) use
($search) {
                                $query-
>where('nama', 'LIKE', "%{$search}%");
                                });
                                })
                                ->count();
    }

    $data = array();
    $data_siswas = $siswas;
    if(!empty($data_siswas))
    {
        foreach ($data_siswas as $index
=> $siswa)
        {
            //Apa yang akan ditampilkan
            di tiap2 row
            $nestedData['nisn'] =
$siswa->nisn;
            $nestedData['nama'] =
$siswa->nama;

            $nestedData['jenis_kelamin'] = $siswa-
>jenis_kelamin;
            $sekolah =
Sekolah::find($siswa->sekolah_id);
            if($siswa->sekolah)
            {
                $sekolah = $siswa-
>sekolah->nama;
            }
            else
            {
                $sekolah = 'Data Sekolah
Siswa Belum Terinput';
            }

```

```

        $nestedData['sekolah_id'] =
$sekolah;
        $nestedData['username'] =
$siswa->username;
        $nestedData['action'] = '<a
href="/siswa/edit/' . $siswa->id . '"><button
class="btn bg-blue edit"><i class="material-
icons">mode_edit</i><span>Edit</span></butt
on></a>
        </button><button
class="btn      bg-red      delete"      data-
toggle="modal"      data-
target="#modalKonfirmasiDelete"
onclick="deleteClickFunction(\'' . $siswa-
>id . '\')"'      ><i      class="material-
icons">delete</i><span>Hapus</span></button
>
        <button class="btn bg-
green detail" data-toggle="modal" data-
target="#deta0ilModal"
onclick=detailFunction(\'' . $siswa-
>id . '\')><i      class="material-
icons">dehaze</i><span>Detail</span></butto
n><button      id="resetPassword"
onclick="resetPassword(\'' . $siswa->id . '\')"'
class="btn btn-default" type="button">Reset
Password</button>';

        //      ID      setter,      untuk
keperluan ajax delete
        $nestedData['DT_RowId'] =
$siswa->id;

        $data[] = $nestedData;

    }

}

//<-- Gak Perlu Diubah -->

```

```

        $json_data = array(
            "draw" =>
intval($request->input('draw')),
            "recordsTotal" =>
intval($totalData),
            "recordsFiltered" =>
intval($totalFiltered),
            "data" =>
$data
        );

        return json_encode($json_data);
        //<-- Gak Perlu Diubah END -->
    }

    public function viewCreate()
    {
        $sekolahs = Sekolah::all();

        return
view('pages.proktor.siswa.create',
compact('sekolahs'));
    }

    public function viewImport()
    {
        return
view('pages.proktor.siswa.import');
    }

    public function viewImportPhoto()
    {
        return
view('pages.proktor.siswa.import_photo');
    }

    public function create()
    {
        return back();
    }

```

```

        $this->validate(request(), [
            'nispn' =>
'required|unique:user,nispn',
            'username' =>
'required|unique:user,username',
            'password' =>
'required|confirmed|min:8|max:40',
            'foto' =>
'mimes:jpeg,png,jpg,gif,svg|max:2048'
        ]);

        $data = request()->all();

        // file upload
        $imasename =
$data['nispn'].'.'.request()->file('foto')-
>getClientOriginalExtension();
        request()->file('foto')-
>move(base_path().
'/public/img/foto_siswa/', $imasename);

        $data['id'] = Uuid::generate()-
>string;
        $data['foto'] = $imasename;
        $data['password'] =
bcrypt($data['password']);
        $data['level'] = 'siswa';
        // dd($data);

        $siswa = User::create($data,
['except' => ['_token', 'submit',
'password_confirmation']]);

        return redirect('/siswa');
    }

    public function viewEditor($siswa_id)

```

```

    {
        $siswa = User::find($siswa_id);
        $sekolahs = Sekolah::all();

        return
view('pages.proktor.siswa.edit',
compact('siswa', 'sekolahs'));
    }

    public function edit()
    {
        $data = request()->all();

        Validator::make($data, [
            'nisn' => ['required',
Rule::unique('user')-
>ignore($data['siswa_id']),],
            'username' => ['required',
Rule::unique('user')-
>ignore($data['siswa_id']),],
            'password' => ['nullable',
'confirmed', 'min:8', 'max:40'],
            'foto' =>
['mimes:jpeg,png,jpg,gif,svg', 'max:2048'],
        ]);

        $siswa =
User::find($data['siswa_id']);

        $siswa->nisn = $data['nisn'];
        $siswa->nama = $data['nama'];
        $siswa->jenis_kelamin =
$data['jenis_kelamin'];
        $siswa->tempat_lahir =
$data['tempat_lahir'];
        $siswa->tanggal_lahir =
$data['tanggal_lahir'];
        $siswa->sekolah_id =
$data['sekolah_id'];
    }

```

```

        $siswa->username =
$data['username'];

        if(isset($data['foto']))
        {
            $imagename =
$data['nisan'].'.'.request()->file('foto')-
>getClientOriginalExtension();
            request()->file('foto')-
>move(base_path().
'/public/img/foto_siswa/', $imagename);
            $siswa->foto = $imagename;
        }

        $siswa->save();

        return redirect('/siswa')-
>with('success', 'Data berhasil diubah');
    }

    public function resetPassword()
    {
        $siswa =
User::find(request('siswa_id'));
        $password =
strtoupper(str_random(6));
        $siswa->password =
bcrypt($password);

        $siswa->save();

        return response()->json($password);
    }

    public function delete()
    {
        $id = request('id');
        $siswa = User::find($id);

```

```

        $siswa->delete();

        return response()->json($id);
    }

    public function getSiswa()
    {
        $id = request('id');
        $siswa = User::find($id);
        $siswa->nama_sekolah = $siswa->sekolah->nama;

        return response()->json($siswa);
    }

    public function uploadExcel()
    {
        return back();

        $this->validate(request(), [
            'excel_import' => 'required'
        ]);

        $filename = 'siswa-import-' . Carbon::now()->toDateTimeString() . '.' . request()->file('excel_import')->getClientOriginalExtension();
        request()->file('excel_import')->move(base_path() . '/public/imports/', $filename);

        $count = 0;
        try
        {
            Excel::load('/public/imports/' . $filename,
            function($reader) use (&$count){

```



```

        foreach($reader->toArray()
as $row)
        {
            $sekolah_id =
Sekolah::where('nama', $row['sekolah_id'])->
>pluck('id')->first();
            $row['sekolah_id'] =
$sekolah_id;
            $row['id'] =
Uuid::generate()->string;
            $row['level'] = 'siswa';
            $row['username'] =
$row['nisan'];
            $row['foto'] =
$row['nisan'].'.jpg';
            $row['password'] =
bcrypt($row['nisan']);

            User::create($row);
            $count++;
        }
    });
} catch (\Exception $e) {

unlink(base_path().'/public/imports/'.$file
name);

        return
redirect('/siswa/import/data')->
>withErrors(['message' => $e->
>getMessage()]);
    }
    return redirect('/siswa')->
>with('success', 'Data sebanyak '.$count.'
siswa berhasil diimpor');
}

public function uploadPhoto()
{
    $zip = request()->file('zip_photo');

```

```

        $original_filename = pathinfo($zip-
>getClientOriginalName(),
PATHINFO_FILENAME);
        $filename = $original_filename.'-
'.Carbon::now()-
>toDateTimeString().'.'.$zip-
>getClientOriginalExtension();
        $path
base_path().'public/imports/'
        $zip->move($path, $filename);
        Zipper::make($path.$filename)-
>extractTo(base_path().'public/img/foto_si
swa/');

        return redirect('/siswa')-
>with('success', 'Foto berhasil diimpor');
    }

    public function updateKeterangan()
    {
        $ujian_siswa
UjianSiswa::find(request('id'));
        $ujian_siswa->keterangan
request('keterangan');
        $ujian_siswa->save();

        return response()->json('ok');
    }

    //print kartu peserta
    public function indexPrintKartuPeserta()
    {
        return
view('pages.proktor.siswa.print_kartu_peser
ta');
    }

    public function showDatatableskp()
    {

```

```

        return
view('pages.proktor.siswa.print_kartu_peserta_datatables');
    }

    public function masukkanKeUjian()
    {
        $sesi_ids =
array_unique(request('sesi'));
        $siswas = User::whereIn('id',
request('siswas'))->get();
        foreach ($sesi_ids as $id_jadwal)
        {
            $jadwal =
JadwalUjian::find($id_jadwal);
            foreach ($siswas as $siswa)
            {
                $data['user_id'] = $siswa->id;
                $data['id'] =
Uuid::generate()->string;
                $data['paket_id'] = "0";
                $data['jadwal_ujian_id'] =
$jadwal->id;
                $data['waktu_mulai'] =
$jadwal->waktu_mulai;
                $data['waktu_selesai'] =
$jadwal->waktu_selesai;
                $ujian_siswa =
UjianSiswa::create($data);
            }
            $this->random($jadwal->id);
        }

        return redirect('/jadwal')->with('success', 'Peserta ujian berhasil diperbaharui');
    }

```

```

public function datatableskip(Request
$request)
{
    //Isi dengan kolom
    $columns = array(
        0 => 'id',
        1 => 'nisn',
        2 => 'nama',
        3
    =>
'jenis_kelamin',
        4 => 'sekolah_id'
    );

    //<-- Gak Perlu Diubah -->
    $totalData =
count(User::where('level', 'siswa')->get());

    $totalFiltered = $totalData;

    $limit = $request->input('length');
    $start = $request->input('start');
    $order = $columns[$request-
>input('order.0.column')];
    $dir = $request-
>input('order.0.dir');

    if(empty($request-
>input('search.value')))
    {
        $siswas =
User::where('level', 'siswa')
            ->offset($start)
            ->limit($limit)
            -
>orderBy($order,$dir)
            ->get();
    }
    else {

```

```

        $search = $request->
>input('search.value');
        //<-- Gak Perlu Diubah END -->
        //Tambah orWhere di kolom yang akan
        dijadikan di search
        $siswas = User::where('level',
'siswa')

-
>where(function ($q) use ($search) {
        $q-
>where('nama', 'LIKE', "%{$search}%")
-
>orWhere('nisn', 'LIKE', "%{$search}%")
-
>orWhereHas('sekolah', function($query) use
($search){
        $query-
>where('nama', 'LIKE', "%{$search}%");
        });
-
>offset($start)
-
        ->limit($limit)
-
>orderBy($order,$dir)
        ->get();
        //Tambah orWhere di kolom yang akan
        dijadikan di search
        $totalFiltered =
User::where('level', 'siswa')
-
>where(function ($q) use ($search) {
        $q-
>where('nama', 'LIKE', "%{$search}%")
-
>orWhere('nisn', 'LIKE', "%{$search}%")
-
>orWhereHas('sekolah', function($query) use
($search){

```

```

$query-
>where('nama', 'LIKE', "%{$search}%");
    });
    })
    ->count();
    }

    $data = array();
    $data_siswas = $siswas;
    if(!empty($data_siswas))
    {
        foreach ($data_siswas as $index
=> $siswa)
        {
            //Apa yang akan ditampilkan
            di tiap2 row
            $nestedData[0] = $siswa->id;
            $nestedData[1] = $siswa->
            >nisn;
            $nestedData[2] = $siswa->
            >nama;
            $nestedData[3] = $siswa->
            >jenis_kelamin;
            $sekolah =
            Sekolah::find($siswa->sekolah_id);
            if($siswa->sekolah_id)
            {
                $sekolah = $siswa->
            >sekolah->nama;
            }
            else
            {
                $sekolah = 'Data Sekolah
            Siswa Belum Terinput';
            }
            $nestedData[4] = $sekolah;

            $data[] = $nestedData;
        }
    }
}

```

```

    }
}

    //<-- Gak Perlu Diubah -->
    $json_data = array(
        "draw" =>
intval($request->input('draw')),
        "recordsTotal" =>
intval($totalData),
        "recordsFiltered" =>
intval($totalFiltered),
        "data" =>
$data
    );

    return json_encode($json_data);
    //<-- Gak Perlu Diubah END -->
}

public function printKartuPeserta()
{
    $siswas = User::whereIn('id',
request('siswas'))->get();
    // dd($siswas);
    $server = Server::first();

    $data['siswas'] = $siswas;
    $data['server'] = $server->nama;
    $pdf =
PDF::loadView('pages.proktor.siswa.print.kar
rtupeserta', $data);
    return $pdf->download('Kartu
Peserta.pdf');
}
//print kartu peserta END

public function hasilUjianPage() {

```

```

        return
view('pages.admin.ujian.index');
    }
    // get details of finished (status == 2)
siswa's ujian
    public function findSiswa($no_peserta) {
        $siswa_details =
User::where('nomer_peserta', $no_peserta)-
>first();
        if (!$siswa_details) {
            return response()-
>json(['message' => 'Not found.'], 404);
        }
        $siswa = $siswa_details;
        $ujian = $siswa_details-
>ujianSiswas->where('status', 2);
        $paket = $siswa_details-
>ujianSiswas[0]->paket->nama;
        $pelajaran = $siswa_details-
>ujianSiswas[0]->paket->pelajaran->nama;

        $sekolah = $siswa_details->sekolah-
>nama;
        // TODO: nilai pilgan from where?
        return response()->json([
            'siswa' => [
                'id' => $siswa->id,
                'nama' => $siswa->nama,
                'no_peserta' => $siswa-
>nomer_peserta
            ],
            'ujian' => $ujian,
            // 'paket' => $paket,
            // 'pelajaran' => $pelajaran
            'sekolah' => $sekolah
        ]);
    }

```



```

public function
detailUjianSiswa($ujian_id) {
    $ujian =
    UjianSiswa::find($ujian_id);
    // get packet n mapel info
    $packet = $ujian->paket->nama;
    $mapel = $ujian->paket->pelajaran-
>nama;
    $nama_siswa = $ujian->user->nama;

    // turn json into array, count
variables are to figure out how many soals
there are
    $jawaban_siswa = json_decode($ujian-
>jawaban_siswa, true);
    $kunjaw = json_decode($ujian-
>kunci_jawaban, true);

    $total_soal_count = count($kunjaw);
    $total_essai_count =
    $total_soal_count - count($jawaban_siswa);

    $soal_order = json_decode($ujian-
>random_soal, true);

    $esai_idx_start = $total_soal_count
- $total_essai_count;

    // get ids of jawaban essai
    $essai_ids =
array_slice($soal_order, $esai_idx_start);

    $jawaban_siswa_essai = array();
    // get jawaban essai of siswa
    for ($i=0; $i < count($essai_ids);
    $i++) {
        $essai =
        JawabanEssai::where('soal_id',
        $essai_ids[$i])

```

```

                                ->where('user_id',
$ujian->user->id)->first();

array_push($jawaban_siswa_essai,      $essai-
>jawaban_siswa);
    }
    // saves pilgan n essay jawaban
    $jawaban_siswa                      =
array_merge($jawaban_siswa,
$jawaban_siswa_essai);

    return
view('pages.admin.ujian.hasil_print',
compact('jawaban_siswa', 'kunjaw', 'packet',
'mapel', 'nama_siswa'));
    }

    public                                function
hasilUjianPerJadwal($jadwal_id) {
    $jadwal                                =
JadwalUjian::find($jadwal_id);
    $schools = Sekolah::all();
    return
view('pages.admin.ujian.index_jadwal',
compact('jadwal', 'schools'));
    }

    public                                function
getHasilUjianJadwal($jadwal_id, $sekolah_id)
{
    // get ujian siswas according to
given jadwal
    try {
        $ujian_siswas                      =
User::select('id', 'nama', 'nomer_peserta')-
>where('sekolah_id',      $sekolah_id)-
>whereHas('ujianSiswas',    function($query)
use($jadwal_id) {

```

```

        $query-
>where('jadwal_ujian_id', $jadwal_id);
        })->with('ujianSiswas')->get();

        $sekolah =
Sekolah::find($sekolah_id);
        $jadwal =
JadwalUjian::find($jadwal_id);
        } catch (Exception $e) {
            Log::info('Error message: ' .
$e->getMessage());
            return redirect()->back();
        }

        // return $sekolah_id;

        return
view('pages.admin.ujian.hasil_sekolah',
compact('ujian_siswas', 'sekolah',
'jadwal'));
    }
}

```

20. Soal Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Uuid;
use Carbon\Carbon;
use App\Models\Soal;
use App\Models\Paket;
use App\Models\Latex;
use App\Models\JadwalUjian;
use App\Models\Settings;

```

```

use PDF;
use App\Helpers\RandomizeHelper;
use Log;
use App\Contracts\IEssayCounter;
use Auth;
use Illuminate\Support\Facades\Config;

class SoalController extends Controller
{
    /* Constructor
       -Only auth
    */
    public function __construct()
    {
        // $this->middleware('auth');
    }

    /* Read
       - Read
       tampilkan soal menurut paket
    */
    public function index($paket_id)
    {
        $paket = Paket::find($paket_id);
        if (!$paket) return
        redirect('paket')->withErrors(['
            message' => 'Paket '.$paket_id.'
            belum dibuat'
        ]);

        // modify soals ordering

        $id = $paket_id;

        $soals_pilgan =
        Soal::where('paket_id',$id)-
        >where('tipe_soal', 'pilihan_ganda')-
        >orderBy('created_at', 'asc')->get()
        ->toArray();
    }
}

```

```

        $soals_essay =
        Soal::where('paket_id',$id)-
        >where('tipe_soal', 'essai')-
        >orderBy('created_at', 'asc')->get()
        ->toArray();

        $soals_pilgan =
        RandomizeHelper::randomizeQuestions($soals_
        pilgan) ['questions'];

        $soals_essay =
        RandomizeHelper::randomizeEssayQuestions($s
        oals_essay,
        count($soals_pilgan) ['questions'];

        $mudah = Soal::where('paket_id',
        $paket_id)->where('tingkat_kesulitan', 1)-
        >count();

        $sedang = Soal::where('paket_id',
        $paket_id)->where('tingkat_kesulitan', 2)-
        >count();

        $sulit = Soal::where('paket_id',
        $paket_id)->where('tingkat_kesulitan', 3)-
        >count();

        $soals = array_merge($soals_pilgan,
        $soals_essay);

        $no = 1;
        return view('pages/guru/soal',
        compact('soals', 'no', 'paket', 'mudah',
        'sedang', 'sulit'));
    }

    public function create(Request
    $request,$paket_id){
        $date = Carbon::now();
        $limit =
        Settings::where('id','batas_input_soal')-
        >first();
        if($date > $limit->value)

```

```

        return back()-
>with('error','Batas waktu sudah lewat!');

        $request->session()-
>put('paket_id',$paket_id);
        $data['paket_id'] = $paket_id;

        if ($request->input('tipe') ==
'essai'){
            $data['min_anchor'] =
Soal::where('paket_id', $paket_id)-
>where('tipe_soal', 'pilihan_ganda')-
>count() + 1;
            $data['max_anchor'] =
Soal::where('paket_id', $paket_id)-
>where('tipe_soal', 'essai')->count();
            return
view('/pages/guru/soal_create_essai',
$data);
        }

        else if ($request->input('tipe') ==
'pilihan_ganda'){
            // max_anchor is used to set
maximum ordering of anchor number
            $data['max_anchor'] =
Soal::where('paket_id', $paket_id)-
>where('tipe_soal', 'pilihan_ganda')-
>count() + 1;
            return
view('/pages/guru/soal_create', $data);
        }

    }

    public function storeUpload(Request
$request)
    {
        $file = request()->file('file');

```

```

        $mime = $file->getMimeType();

        if(strpos($mime, "image/"))
        {
            $filename = 'img-soal-'
            '.Uuid::generate()->string.'.'.$file-
            >getClientOriginalExtension();
            $file-
            >move(base_path().'public/imports/soal_med
            ia/'.'.$request->session()-
            >get('paket_id').'images/', $filename);
            // $file->move(base_path().
            '/public/img/foto_siswa/', $imagename);
            $file_path =
            '/imports/soal_media/'.'.$request->session()-
            >get('paket_id').'images/'.'.$filename;
            $file_type = 'image';
        }
        else if(strpos($mime, "video/"))
        {
            $filename = 'video-soal-'
            '.Uuid::generate()->string.'.'.$file-
            >getClientOriginalExtension();
            $file-
            >move(base_path().'public/imports/soal_med
            ia/'.'.$request->session()-
            >get('paket_id').'videos/', $filename);
            // $file->move(base_path().
            '/public/img/foto_siswa/', $imagename);
            $file_path =
            '/imports/soal_media/'.'.$request->session()-
            >get('paket_id').'videos/'.'.$filename;
            $file_type = 'video';
        }
        else if(strpos($mime, "audio/"))
        {

```

```

        $filename = 'audio-soal-
        '.Uuid::generate()->string.'.'.$file-
>getClientOriginalExtension();
        $file-
>move(base_path().'public/imports/soal_med
ia/'.'.$request->session()-
>get('paket_id').'/audios/', $filename);
        // $file->move(base_path().
'/public/img/foto_siswa/', $imagename);
        $file_path =
'/imports/soal_media/'.'.$request->session()-
>get('paket_id').'/audio/'.'.$filename;
        $file_type = 'audio';
    }

    $data = array(
        'file_path' => $file_path,
        'file_name' => $filename,
        'file_type' => $file_type,
        'status' => 200
    );

    return response()->json($data);
}

public function deleteUpload()
{
    $filename = request('filename');

    if(strstr($filename, 'img-soal'))
    {

unlink(base_path().'public/imports/soal_me
dia/'.'.$request->session()-
>get('paket_id').'/images/'.'.$filename);
    }
    else if(strstr($filename, 'audio-
soal'))

```



```

    {

unlink(base_path().'/public/imports/soal_media/' . $request->session()-
>get('paket_id').'/audios/' . $filename);
    }
    else if(strpos($filename, 'video-
soal'))
    {

unlink(base_path().'/public/imports/soal_media/' . $request->session()-
>get('paket_id').'/videos/' . $filename);
    }

    return response()->json(200);
}

public function viewUpload()
{
    return
view('/pages/guru/image_uploader/_image-
dialog');
}

public function updateBobot($paket_id)
{
    $paket = Paket::find($paket_id);
    $paket->bobot_pilihan_ganda =
request('bobot_pilihan');
    $paket->save();

    return back();
}

private function insertLatex($text,
$soal_id)
{

```

```

        Latex::where('Soal_id',$soal_id)-
>delete();
        $arrayLatex = [];
        $textReplace = str_replace('<img
title=', '<latex>', $text);
        $textReplace = str_replace('
/>', '</latex>', $textReplace);
        $posisiAwal = strpos($textReplace,
'<latex>');
        if($posisiAwal!==false){
            while (1) {
                $posisiAkhir =
strpos($textReplace,
'</latex>', $posisiAwal);
                $latex =
substr($textReplace, $posisiAwal+7,
$posisiAkhir-($posisiAwal+7));
                array_push($arrayLatex,
$latex);
                $posisiAwal =
strpos($textReplace, '<latex>',
$posisiAwal+8);
                if($posisiAwal===false)
break;
            }
        }
        foreach ($arrayLatex as $value) {
            $domCodeCogs = '<img
title= '.$value.' />';
            $domMathjax = '<span
class="math-
tex">\(' . explode(' ', $value)[1] . '\)</span>'
;
            $newLatex = new Latex();
            $newLatex->Soal_id = $soal_id;
            $newLatex->latex_codecogs =
$domCodeCogs;
            $newLatex->latex_mathjax =
$domMathjax;

```

```

        $newLatex->save();
    }
}

public function viewEditor($soal_id)
{
    $date = Carbon::now();
    $limit =
Settings::where('id','batas_input_soal')-
>first();
    if($date > $limit->value)
        return back()-
>with('error','Batas waktu sudah lewat!');

    $soal = Soal::find($soal_id);

    if (request('tipe') == 'essai'){
        $min_anchor =
        Soal::where('paket_id', $soal->paket_id)-
        >where('tipe_soal', 'pilihan_ganda')-
        >count() + 1;
        $max_anchor =
        Soal::where('tipe_soal', 'essai')-
        >where('paket_id', $soal->paket_id)-
        >count();
        return
        view('pages/guru/soal_edit_essai',
        compact('soal', 'min_anchor',
        'max_anchor'));
    }
    else if (request('tipe') ==
    'pilihan_ganda'){
        $max_anchor =
        Soal::where('tipe_soal', 'pilihan_ganda')-
        >where('paket_id', $soal->paket_id)-
        >count();
        return
        view('pages/guru/soal_edit', compact('soal',
        'max_anchor'));
    }
}

```

```

    }

}

/* Edit
- Edit
    tampilan editing nya pake modal
*/
public function edit()
{
    $data = request()->all();

    // check for duplicates
    if (isset($data['anchor_number']) &&
    $data['anchor_number']) {
        // check anchor number - cannot
        be more than available questions
        $is_over_max =
        Soal::where('paket_id', $data['paket_id'])-
        >count() < $data['anchor_number'] ? true :
        false;

        if ($is_over_max) {
            return redirect()->back()-
            >withInput()->withErrors('Nomor
            urutan
            anchor melebihi jumlah soal yang ada. Mohon
            pilih nomor yang lebih kecil.');
```

```

        if ($duplicate) {
            return redirect() -
>back()->withInput()->withErrors('Sudah ada
nomor urutan untuk soal ini. Mohon untuk
pilih nomor urutan yang berbeda');
        }

    }

    if(is_null($data['anchor_number']))
    {
        $anchor_number = 0;
    } else {
        $anchor_number =
$data['anchor_number'];
    }

    $soal =
Soal::find($data['soal_id']);

    $soal->deskripsi = $data['deskripsi'];
    $soal->tingkat_kesulitan =
$data['tingkat_kesulitan'];
    if(isset($data['level_kognitif']))
        $soal->level_kognitif =
$data['level_kognitif'];
    $soal->kunci_jawaban =
$data['kunci_jawaban'];
    $soal->anchor_number = $anchor_number;

    if ($soal-> tipe_soal ==
'pilihan_ganda')
    {
        $soal->pilihan_a =
$data['pilihan_a'];
        $soal->pilihan_b =
$data['pilihan_b'];
        $soal->pilihan_c =
$data['pilihan_c'];
    }

```

```

        $soal->pilihan_d =
$data['pilihan_d'];
        $soal->pilihan_e =
$data['pilihan_e'];
        $soal->kunci_jawaban =
$data['kunci_jawaban'];

        if($soal->save())
        {
            $this->
>insertLatex($data['deskripsi'].$data['pili
han_a'].$data['pilihan_b'].$data['pilihan_c
'].$data['pilihan_d'].$data['pilihan_e'],
$data['soal_id']);
        }
    }
    else if ($soal->tipe_soal ==
'essai')
    {
        // $soal->bobot_essai =
$data['bobot_essai'];
        if($soal->save())
        {
            $this->
>insertLatex($data['deskripsi'],
$data['soal_id']);
        }
    }

    $soal->save();

    return
redirect('/soal/'.$data['paket_id']);
}

// function to store new soal
public function store(Request $request,
IEssayCounter $ec)
{

```

```

        $data = request()->all();
        $data['id'] = Uuid::generate()-
>string;

        if ($request->anchor_number) {
            // check anchor number - cannot
            be more than available questions
            $is_over_max =
            Soal::where('paket_id', $request->paket_id)-
            >count() + 1 < $request->anchor_number ?
            true : false;

            if ($is_over_max) {
                return redirect()->back()-
            >withInput()->withErrors('Nomor urutan
            anchor melebihi jumlah soal yang ada. Mohon
            pilih nomor yang lebih kecil.');
```

```

            }

            // check for duplicates
            $duplicate =
            Soal::where('paket_id', $request->paket_id)
            ->where('anchor_number',
            $request->anchor_number)
            ->count();
```

```

            if ($duplicate) {
                return redirect()-
            >back()->withInput()->withErrors('Sudah ada
            nomor urutan untuk soal ini. Mohon untuk
            pilih nomor urutan yang berbeda');
```

```

            }

            if(is_null($data['anchor_number']))
            {
                $data['anchor_number'] = 0;
            }
        }
    }
}
```

```

        $soal = Soal::create($data,
['except' => ['_token', 'submit']]);

        if($data['tipe_soal'] == "esai")
            $ec->initSoalCounter($soal-
>id);

        return
redirect('/soal/' . $data['paket_id']);
    }

    public function delete()
    {
        $date = Carbon::now();
        $limit =
Settings::where('id', 'batas_input_soal')-
>first();
        if($date > $limit->value)
            return response()->json(['error'
=> 'Batas waktu sudah lewat!']);

        $id = request('id');
        $soal = Soal::find($id);
        $soal->delete();
        Latex::where('Soal_id', $id)-
>delete();
        return response()->json($id);
    }

    public function getSoal()
    {
        $id = request('id');
        $soal = Soal::find($id);

        return response()->json($soal);
    }

    public function getAllSoal()
    {

```



```

        $id = request('id');
        try {
            $soals_pilgan =
Soal::where('paket_id',$id)-
>where('tipe_soal',          'pilihan_ganda')-
>orderBy('created_at', 'asc')->get()
            ->toArray();

            $soals =
RandomizeHelper::randomizeQuestions($soals_
pilgan) ['questions'];

            $soals_essay =
Soal::where('paket_id',$id)-
>where('tipe_soal',          'essai')-
>orderBy('created_at', 'asc')->get()
            ->toArray();

            $soals2 =
RandomizeHelper::randomizeEssayQuestions($s
oals_essay,
count($soals_pilgan) ['questions']);
        } catch (Exception $e) {
            Log::info('Error message: ' .
$e->getMessage());
            return;
        }

        $arr_soals = array_merge($soals,
$soals2);

        return response()->json($arr_soals);
    }

    public function printSoal()
    {
        $user_offline =
Config::get('offline.id_user');

```

```

        if      (!in_array(Auth::user()->id,
$user_offline) || Auth::user()->level ==
"admin")) {
            $text = "Alamat IP telah
tersimpan";
            return $text;
        }
        $id = request('id');
        $packet = Paket::find($id);
        if (Auth::user()->id != '411b9720-
4128-11ea-b496-0b1bcee8dc25' &&
Auth::user()->level == "admin" && $packet->
nama[-1]!="5") {
            return redirect()->back()-
>with('error', 'Akses ditolak');
        }
        $jadwal =
JadwalUjian::where('pelajaran_id', $packet->
pelajaran_id)->where('deskripsi', 'not
like', "%|OFFLINE|%")->where('deskripsi',
'not like', "%|SUSULAN|%")->first();

        try {
            $soals_pilgan =
Soal::where('paket_id',$id)-
>where('tipe_soal', 'pilihan_ganda')-
>orderBy('created_at', 'asc')->get()
            ->toArray();

            $soals =
RandomizeHelper::randomizeQuestions($soals_
pilgan) ['questions'];

            $soals_essay =
Soal::where('paket_id',$id)-
>where('tipe_soal', 'essai')-
>orderBy('created_at', 'asc')->get()
            ->toArray();

```

```

        $soals2 =
RandomizeHelper::randomizeEssayQuestions($s
oals_essay,
count($soals_pilgan))['questions'];
    } catch (Exception $e) {
        Log::info('Error message: ' .
$e->getMessage());
        return;
    }

    $arr_soals = array_merge($soals,
$soals2);

    return view('pages.guru.soal_print',
['soals' => $arr_soals, 'packet' =>
$packet]);
    }

    public function hapusPilihan()
    {
        $id = request('id');
        $soal = Soal::find($id);
        $soal->pilihan_e = null;
        $soal->save();

        return response()->json($id);
    }
    // find if any chosen anchor_number has
been set
    public function
checkAnchorNumberDuplicates(Request
$request, $packet_id) {
        try {
            // for new soal
            if ($request->type == "create")
            {
                $duplicate =
Soal::where('paket_id', $packet_id)

```

```

        ->where('anchor_number',
$request->anchor_number)
        ->count();
    } else {
        // for edited soal, dont
count current soal
        $duplicate =
Soal::where('paket_id', $packet_id)
        ->where('anchor_number',
$request->anchor_number)
        ->where('id', '!= ',
$request->id)
        ->count();
    }

    } catch (Exception $e) {
        return response()-
>json(['message' => $e->getMessage()], 500);
    }

    return response()->json(['message'
=> $duplicate]);
}
}

```

21. Ujian Controller

```

<?php

namespace App\Http\Controllers;

use Carbon\Carbon;
use Uuid;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Validator;
use App\Models\User;

```

```

use App\Models\Paket;
use App\Models\Soal;
use App\Models\Latex;
use App\Models\UjianSiswa;
use App\Models\JadwalUjian;
use App\Models\JawabanEssai;
use App\Models\Server;
use App\Models\Sekolah;

class UjianController extends Controller
{
    public function login(Request $request){
        if(Auth::attempt(['username' =>
$request->input('username'),
        'password' => $request-
>input('password')))) {
            // Auth::user()->ujiansiswa-
>status=1;
            $ujiansiswas =
UjianSiswa::where('user_id', Auth::user()-
>id)->get();
            if(count($ujiansiswas))
            {
                $ujiansiswa = $ujiansiswas-
>first();
                $ujiansiswa->status=1;
                $ujiansiswa->save();
            }
            return redirect()-
>action('UjianController@dataPeserta');
        }
        return redirect('/')-
>with('error','Username atau Password
salah');
    }

    public function index(){
        return view('welcome');
    }
}

```

```

public function serverName()
{
    $server = new Server;
    $server->nama = request('server');
    $server->save();

    $sekolah = Sekolah::first();
    $sekolah->nama = request('sekolah');
    $sekolah->save();

    return redirect('/');
}

public function
getSoal($ujian_siswa_id){
    $user = Auth::user();

    $ujian_siswa =
    UjianSiswa::find($ujian_siswa_id);
    $paket =
    Paket::where('id',$ujian_siswa->paket_id)-
    >first();

    $random_soal =
    json_decode($ujian_siswa->random_soal);
    $random_soal_ori = $random_soal;
    $jumlah_soal = sizeof($random_soal);
    $random_soal = implode('\','\'',
    $random_soal);
    $random_soal =
    substr_replace($random_soal, '\'', 0, 0);
    $random_soal =
    substr_replace($random_soal,
    '\'',
    strlen($random_soal), 0);

    $random_jawaban =
    json_decode($ujian_siswa->random_jawaban);
    // dd($random_jawaban);

```

```

        $soals =
Soal::where('paket_id', $paket->id)-
>orderByRaw("\DB::raw("FIELD(id,
$random_soal)"))->paginate(1);
        foreach ($soals as $soal) {
            $latexs =
Latex::where('soal_id', $soal->id->get());
            foreach ($latexs as $key =>
$valueLatex) {
                $soal->deskripsi =
str_replace($valueLatex->latex_codecogs,
$valueLatex->latex_mathjax, $soal-
>deskripsi);
                $soal->pilihan_a =
str_replace($valueLatex->latex_codecogs,
$valueLatex->latex_mathjax, $soal-
>pilihan_a);
                $soal->pilihan_b =
str_replace($valueLatex->latex_codecogs,
$valueLatex->latex_mathjax, $soal-
>pilihan_b);
                $soal->pilihan_c =
str_replace($valueLatex->latex_codecogs,
$valueLatex->latex_mathjax, $soal-
>pilihan_c);
                $soal->pilihan_d =
str_replace($valueLatex->latex_codecogs,
$valueLatex->latex_mathjax, $soal-
>pilihan_d);
                $soal->pilihan_e =
str_replace($valueLatex->latex_codecogs,
$valueLatex->latex_mathjax, $soal-
>pilihan_e);
            }
        }
        $soals-
>withPath("$ujian_siswa_id");

```

```

        $soals->random_jawaban =
$random_jawaban;

        $jawabans = $ujian_siswa-
>jawabansiswa;

        $jawabans = json_decode($jawabans);

        $jumlah_esai = $jumlah_soal -
sizeof($jawabans);
        $jumlah_pilgan = sizeof($jawabans);
        $jawabans_essai = array();
        for ($i=0; $i < $jumlah_esai; $i++)
        {
            $soal_id =
$random_soal_ori[$jumlah_pilgan - $i];
            $jawaban_essai =
JawabanEssai::where('soal_id',$soal_id)-
>where('ujian_siswa_id',$ujian_siswa_id)-
>first();

            $hasil = '';
            if($jawaban_essai) {
                $len =
strlen($jawaban_essai->jawaban_siswa);
                if($jawaban_essai-
>jawaban_siswa[0]=='*' && $jawaban_essai-
>jawaban_siswa[$len-1]=='*')
                    $hasil = '*';
                else $hasil = 'O';
            } else {
                $hasil = ' ';
            }

            array_push($jawabans_essai,
$hasil);
        }
        // dd($jawabans_essai);

```



```

        $waktu_mulai =
Carbon::parse($ujian_siswa->jadwalUjian-
>waktu_mulai,'Asia/Jakarta')-
>timestamp*1000;
        $waktu_selesai =
Carbon::parse($ujian_siswa->jadwalUjian-
>waktu_selesai,'Asia/Jakarta')-
>timestamp*1000;
        $waktu_sekarang =
Carbon::now('Asia/Jakarta')-
>timestamp*1000;

        $data['soals'] = $soals;
        $data['ujian_siswa_id'] =
$ujian_siswa_id;
        $data['waktu_mulai'] = $waktu_mulai;
        $data['waktu_sekarang'] =
$waktu_sekarang;
        $data['waktu_selesai'] =
$waktu_selesai;
        $data['jawabans'] = $jawabans;
        $data['jumlah_pilgan'] =
$jumlah_pilgan;
        $data['jawabans_essai'] =
$jawabans_essai;

        return view('pages.siswa.soal',
$data);
    }

    public function postJawaban() {
        if(request('tipe_soal')=='essai') {
            $jawaban = request('jawaban');
            $ujian_siswa_id = request('id');
            $soal_id = request('soal_id');
            $no = request('no')-1;

            $jawaban_essai =
JawabanEssai::where('soal_id',$soal_id)-

```

```

>where('ujian_siswa_id',$ujian_siswa_id)-
>first();
        if($jawaban_essai) {
            // dd($jawaban_essai);
            $jawaban_essai-
>jawaban_siswa = $jawaban;
            $jawaban_essai->save();
        } else {
            // dd($soal_id);
            $jawaban_essai      =      new
JawabanEssai();
            $jawaban_essai->id      =
Uuid::generate();
            $jawaban_essai->soal_id =
$soal_id;
            $jawaban_essai-
>ujian_siswa_id = $ujian_siswa_id;
            $jawaban_essai-
>jawaban_siswa = $jawaban;
            $jawaban_essai->save();
        }
        return      redirect()->back() -
>with('show', 'Sukses Menyimpan Jawaban');
    }
    else {
        $jawaban = request('jawaban');
        $ujian_siswa_id = request('id');
        $no = request('no')-1;

        $ujian_siswa      =
UjianSiswa::where('id',$ujian_siswa_id)-
>first();
        $jawabans      =
json_decode($ujian_siswa->jawaban_siswa);
        if($jawaban == null)
            $jawabans[$no] = " ";
        else
            $jawabans[$no] = $jawaban;
    }
}

```

```

        $ujian_siswa->jawaban_siswa =
json_encode($jawabans);
        $ujian_siswa->save();
    }
}

public function getJawaban() {
    $ujian_siswa_id = request('id');
    $tipe_soal = request('tipe_soal');
    // return response()-
>json(request('soal_id'));
    if($tipe_soal == 'esai') {
        $soal_id = request('soal_id');
        $jawaban_esai =
JawabanEssai::where('soal_id',$soal_id)-
>where('ujian_siswa_id',$ujian_siswa_id)-
>first();
        $str = $jawaban_esai-
>jawaban_siswa;
        $len = strlen($str);
        if($str[0] == '*' && $str[$len-
1]=='*') {
            $str = substr($str, 1, -1);
            return response()->json([
                'jawaban' => $str,
                'status' => '*'
            ]);
        }
        return response()->json([
            'jawaban' => $str,
            'status' => ''
        ]);
    } else {
        $no = request('no')-1;

        $ujian_siswa =
UjianSiswa::where('id',$ujian_siswa_id)-
>first();
        // dd($ujian_siswa);
    }
}

```

```

        //      return      response() -
>json($ujian_siswa_id);
        $jawabans =
json_decode($ujian_siswa->jawaban_siswa);
        $jawaban = $jawabans[$no];

        return      response() -
>json($jawaban);
    }

    public function finish(){
        $user = Auth::user();

        return
view('pages.siswa.finish',compact('user'));
    }

    public function dataBelum(){
        return
view('pages.siswa.konfirmasi.data-belum');
    }

    public function dataPeserta(){
        $user = Auth::user();

        return
view('pages.siswa.konfirmasi.data-peserta',
compact('user'));
    }

    public function dataTes(){
        $user = Auth::user();
        $tanggal_sekarang =
Carbon::now('Asia/Jakarta');
        $ujian_siswa =
UjianSiswa::where('user_id',$user->id)-
>where('waktu_selesai','>',$tanggal_sekaran
g)->orderBy('waktu_mulai','asc')->first();

```

```

        if($ujian_siswa == null)
        {
            return
view('pages.siswa.konfirmasi.data-tes-
tidakada',compact('user','ujian_siswa'));
        }
        $jadwal_ujian =
JadwalUjian::where('id',$ujian_siswa-
>jadwal_ujian_id)->first();

        setlocale(LC_TIME, 'id_ID.utf8');
        $tanggal_tes_mulai =
Carbon::parse($jadwal_ujian->waktu_mulai,
'Asia/Jakarta');

        $tanggal_tes_selesai=Carbon::parse($jadwal_ujian->waktu_selesai, 'Asia/Jakarta');
        $waktu_mulai = $tanggal_tes_mulai-
>timestamp*1000;
        $waktu_sekarang =
Carbon::now('Asia/Jakarta')-
>timestamp*1000;

        return
view('pages.siswa.konfirmasi.data-tes',
compact('user','ujian_siswa','jadwal_ujian',
'tanggal_tes_mulai','tanggal_tes_selesai',
'tanggal_sekarang','waktu_sekarang',
'waktu_mulai'));
    }

    public function konfirmasiUjian(){
        $user =
Auth::user();$tanggal_sekarang =
Carbon::now('Asia/Jakarta');
        $ujian_siswa =
UjianSiswa::where('user_id',$user->id)-
>where('waktu_selesai','>',$tanggal_sekarang)-
>orderBy('waktu_mulai','asc')->first();
    }

```

```

        $random_soal =
json_decode($ujian_siswa->random_soal);
        $random_soal_ori = $random_soal;
        $jumlah_soal = sizeof($random_soal);

        $jawabans = $ujian_siswa-
>jawaban_siswa;
        $jawabans = json_decode($jawabans);
        $jumlah_esai = $jumlah_soal -
sizeof($jawabans);
        $jumlah_pilgan = sizeof($jawabans);
        $jawabans_essai = array();
        for ($i=0; $i < $jumlah_esai; $i++)
        {
                $soal_id =
$random_soal_ori[$jumlah_pilgan - $i];
                $jawaban_essai =
JawabanEssai::where('soal_id',$soal_id)-
>where('ujian_siswa_id',$ujian_siswa->id)-
>first();

                $hasil = '';
                if($jawaban_essai) {
                        $jawaban_essai =
JawabanEssai::where('soal_id',$soal_id)-
>where('ujian_siswa_id',$ujian_siswa->id)-
>first();
                        $str = $jawaban_essai-
>jawaban_siswa;
                        $len = strlen($str);
                        if($str[0] == '*' &&
$str[$len-1]=='*') {
                                $str = substr($str, 1, -
1);
                        }
                        $hasil = $str;
                } else {
                        $hasil = ' ';

```

```

    }

    array_push($jawabans_essai,
$hasil);
    }

    $data['jawabans'] = $jawabans;
    $data['jawabans_essai'] =
$jawabans_essai;

    return view('pages.siswa.soal-
component.logout-soal', $data);
    }

    public function gantiPassword(Request
$request)
    {
        $validator =
Validator::make($request->all(), [
            'password' => 'min:8',
        ]);
        if($validator->fails()) {
            return redirect('/ujian/data-
peserta')->with('error', 'Password Minimal 8
Karakter');
        }
        else
        {
            if($request->password !=
$request->confirm_password)
            {
                return
redirect('/ujian/data-peserta')->
with('error', 'Password Konfirmasi Tidak
Cocok');
            }
            else
            {

```

```

        $siswa =
User::find(Auth::user()->id);
        $data['password'] =
bcrypt($request->password);
        $siswa->update($data);
        return
redirect('/ujian/data-peserta')-
>with('success', 'Password Berhasil
Dirubah');
    }
}
}
}

```

22. User Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;
use App\Models\Sekolah;
use App\Models\Pelajaran;
use App\Http\Requests\Admin\UserRequest;
use Uuid;

class UserController extends Controller
{
    public function index()
    {
        $data['users'] =
User::where('level', '!=', 'siswa')-
>paginate(5);
        $data['paginate'] = 5;
        return
view('pages.admin.user.index', $data);
    }
}

```



```

    }

    public function create()
    {
        $data['sekolahs'] = Sekolah::all();
        $data['pelajarans'] = Pelajaran::all();
        return view('pages.admin.user.create', $data);
    }

    public function store(Request $request)
    {
        $user = new User();
        $user->id = Uuid::generate();
        $user->nama = $request->nama;
        $user->foto = $request->foto;
        $user->jenis_kelamin = $request->jenis_kelamin;
        $user->tempat_lahir = $request->tempat_lahir;
        $user->tanggal_lahir = $request->tanggal_lahir;
        $user->nisn = $request->nisn;
        $user->username = $request->username;
        $user->password = bcrypt($request->password);
        $user->level = $request->level;
        $user->sekolah_id = $request->sekolah_id;
        $user->pelajaran_id = $request->pelajaran_id;
        $user->save();
        return redirect('/users');
    }

    public function show($paginate)
    {

```

```

        $data['users'] =
User::paginate($paginate);
        $data['paginate'] = $paginate;
        return
view('pages.admin.user.index', $data);
    }

    public function edit($id)
    {
        $data['user'] = User::find($id);
        $data['sekolahs'] = Sekolah::all();
        $data['pelajarans'] =
Pelajaran::all();

        return view('pages.admin.user.edit',
$data);
    }

    public function update($id, UserRequest
$request)
    {
        $user = User::find($id);
        $user->id = Uuid::generate();
        $user->nama = $request->nama;
        $user->foto = $request->foto;
        $user->jenis_kelamin = $request-
>jenis_kelamin;
        $user->tempat_lahir = $request-
>tempat_lahir;
        $user->tanggal_lahir = $request-
>tanggal_lahir;
        $user->nisn = $request->nisn;
        $user->username = $request-
>username;
        $user->password = bcrypt($request-
>password);
        $user->level = $request->level;
        $user->pelajaran_id = $request-
>pelajaran_id;

```

```

        $user->sekolah_id = $request->sekolah_id;
        $user->save();

        return redirect()->route('users.index')->with('flash_success', 'User berhasil diperbarui');
    }

    public function destroy($id)
    {
        $user = User::find($id);
        $user->delete();
        return redirect('/users');
    }

    public function datatables(Request $request)
    {
        //Isi dengan kolom
        $columns = array(
            0 => 'nama',
            1 => 'username',
            2 => 'level',
            3 => 'action'
        );

        //<-- Gak Perlu Diubah -->
        $totalData = count(User::where('level', '!=', 'siswa'))->get();

        $totalFiltered = $totalData;

        $limit = $request->input('length');
        $start = $request->input('start');
        $order = $columns[$request->input('order.0.column')];
    }

```

```

        $dir = $request->
>input('order.0.dir');

        if (empty($request->
>input('search.value'))) {
            $users = User::where('level',
'!=', 'siswa')
                ->offset($start)
                ->limit($limit)
                ->orderBy($order, $dir)
                ->get();
        } else {
            $search = $request->
>input('search.value');
            //<-- Gak Perlu Diubah END -->
            //Tambah orWhere di kolom yang
            akan dijadikan di search
            $users = User::where('level',
'!=', 'siswa')
                ->where(function ($q) use
($search) {
                    $q->where('nama',
'LIKE', "%{$search}%")
                    -
>orWhere('username', 'LIKE', "%{$search}%")
                    ->orWhere('level',
'LIKE', "%{$search}%")
                    -
>orWhereHas('sekolah', function ($query) use
($search) {
                        $query->
>where('nama', 'LIKE', "%{$search}%");
                    });
                })
                ->offset($start)
                ->limit($limit)
                ->orderBy($order, $dir)
                ->get();
        }
    }
}

```

```

//Tambah orWhere di kolom yang
akan dijadikan di search
$totalFiltered =
User::where('level', 'siswa')
->where(function ($q) use
($search) {
    $q->where('nama',
'LIKE', "%{$search}%")
-
>orWhere('username', 'LIKE', "%{$search}%")
->orWhere('level',
'LIKE', "%{$search}%")
-
>orWhereHas('sekolah', function ($query) use
($search) {
    $query-
>where('nama', 'LIKE', "%{$search}%");
    });
    })
->count();
}

$data = array();
if (!empty($users)) {
    foreach ($users as $user) {
        //Apa yang akan ditampilkan
di tiap2 row
        $nestedData['nama'] = $user-
>nama;
        $sekolah =
Sekolah::find($user->sekolah_id);
        if ($user->sekolah) {
            $sekolah = $user-
>sekolah->nama;
        } else {
            $sekolah = 'Data Sekolah
Siswa Belum Terinput';
        }
    }
}

```

```

        $nestedData['sekolah'] =
$sekolah;
        $nestedData['username'] =
$user->username;
        $nestedData['level'] =
$user->level;
        $nestedData['action'] = '<a
href="/users/' . $user->id . '/edit"><button
class="btn bg-blue edit"><i class="material-
icons">mode_edit</i><span>Edit</span></butt
on></a>
                                </button><button
class="btn      bg-red      delete"      data-
toggle="modal"      data-target="#ModalDelete"
onclick="deleteClickFunction(\'\' . $user-
>username . '\', \'\' . $user->id . '\')' ><i
class="material-
icons">delete</i><span>Hapus</span></button
>';

                                //      ID      setter,      untuk
keperluan ajax delete
        $nestedData['DT_RowId'] =
$user->id;

        $data[] = $nestedData;
    }
}

//<-- Gak Perlu Diubah -->
$json_data = array(
    "draw" =>
intval($request->input('draw')),
    "recordsTotal" =>
intval($totalData),
    "recordsFiltered" =>
intval($totalFiltered),
    "data" => $data
);

```

```

        return json_encode($json_data);
        //<-- Gak Perlu Diubah END -->
    }
}

```

23. Web Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Excel;
use App\Models\Settings;

class WebController extends Controller
{
    public function timeline()
    {
        $jadwal = Settings::find('jadwal');
        $data['jenis'] =
Settings::find('jenis')->value;
        if($jadwal){
            $data['jadwal'] = $jadwal->value;
        }else{
            $data['jadwal'] = '';
        }
        return view('timeline',$data);
    }

    public function vhd()
    {
        $vhmlink1 =
Settings::find('vhmlink1');
        if ($vhmlink1 == NULL) {
            $vhmlink1 = '';

```

```

    } else {
        $vhdlLink1 = $vhdlLink1->value;
    }
    $vhdlLink2
Settings::find('vhdlLink2');
    if ($vhdlLink2 == NULL) {
        $vhdlLink2 = '';
    } else {
        $vhdlLink2 = $vhdlLink2->value;
    }
    $vhdlLink3
Settings::find('vhdlLink3');
    if ($vhdlLink3 == NULL) {
        $vhdlLink3 = '';
    } else {
        $vhdlLink3 = $vhdlLink3->value;
    }
    return view('pages.proktor.vhd',
compact('vhdlLink1','vhdlLink2','vhdlLink3') );
    }
}

```